

Tartu Ülikool  
Loodus- ja täppiseaduste valdkond  
Matemaatika ja statistika instituut

Kristin Avans

**Segmenteerimine ja parameetrite hindamine  
varjatud Markovi mudeli korral**

Matemaatika eriala  
Bakalaureusetöö (9 EAP)

Juhendaja: Jüri Lember

Tartu 2019

# Segmenteerimine ja parameetrite hindamine varjatud Markovi mudeli korral

Bakalaureusetöö  
Kristin Avans

**Lühikokkuvõte.** Käesolevas bakalaureusetöös kirjeldatakse varjatud Markovi mudelit ning meetodeid varjatud Markovi ahela realisatsiooni leidmiseks ja parameetrite hindamiseks. Kuna mudelis on varjatud andmed, siis tuleb parameetrite hindamiseks kasutada iteratiivseid algoritme. Samuti võrreldakse omavahel erinevaid hindamismeetodeid ning uuritakse väljundi sõltuvust algühenditest. Tuuakse näiteid erinevate meetodite tööst.

**CERCS teaduseriala:** P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

**Märksõnad:** juhuslikud protsessid, Markovi ahelad, Viterbi algoritm, EM algoritm.

# Segmentation and estimation of the parameters of hidden Markov model

Bachelor's thesis  
Kristin Avans

**Abstract.** In this Bachelor's thesis we describe hidden Markov model and methods to find the realization of the hidden Markov chain and the estimates of unknown parameters. As there are latent variables in the model, we use iterative algorithms to find the estimates of the parameters. We also compare different estimation methods and investigate how outputs depend on initial choice of parameters. We give examples of different methods.

**CERCS research specialisation:** P160 Statistics, operations research, programming, actuarial mathematics.

**Keywords:** stochastic processes, Markov chains, Viterbi algorithm, EM algorithm.

# Sisukord

<b>Sissejuhatus</b>	<b>4</b>
<b>1 Varjatud Markovi mudel</b>	<b>5</b>
1.1 Diskreetse ajaga Markovi ahelad . . . . .	5
1.2 Varjatud Markovi mudel . . . . .	6
1.3 <i>Forward</i> ja <i>backward</i> muutujad . . . . .	9
<b>2 Segmenteerimine</b>	<b>14</b>
2.1 Viterbi algoritm . . . . .	15
2.2 PMAP joondus . . . . .	19
<b>3 Mudeli parameetrite hindamine</b>	<b>22</b>
3.1 Viterbi treening . . . . .	22
3.2 EM algoritm . . . . .	30
3.3 PMAP treening ning hindamismeetodite võrdlus . . . . .	37
<b>Viited</b>	<b>44</b>
<b>4 Lisad</b>	<b>45</b>
4.1 Lisa 1. Simulatsioonides kasutatud R koodid . . . . .	45
4.2 Lisa 2. Alglähenditeks antud parameetrid . . . . .	52

## Sissejuhatus

Markovi ahelad on juhuslikud protsessid, mis kirjeldavad seisundite üleminekuid ajas. Markovi ahelaid iseloomustab omadus, et fikseeritud oleviku korral tulevik ei sõltu minevikust. Praktikas võib tihti ette tulla, et me ei saa otseselt vaadelda Markovi ahelat, vaid näeme seda läbi teise juhusliku protsessi, mida nimetatakse varjatud Markovi protsessiks. Varjatud Markovi mudeliks nimetatakse nende kahe juhusliku protsessi paari. Varjatud Markovi mudelil on rakendusi näiteks bioinformaatikas [1], kõnetuvastuses [2].

Käosoleva bakalaureusetöö eesmärk on anda ülevaade varjatud Markovi mudelist ning kirjeldada, mis meetoditega saab lahendada mudeliga seotud põhiküsimusi: kuidas leida optimaalne varjatud Markovi ahela realisatsioon ning kuidas efektiivselt hinnata mudeli parameetrite väärtusi. Töös keskendutakse lihtsuse mõttes juhule, kus vaatlusi on lõplik arv ja saame vaadata diskreetset tõenäosusjaotust. Samad tulemused kehtivad ka juhul, kus vaatlusi on loenduv arv ning emissiooni-jaotused on pidevad. Pideval juhul tuleb tõenäosusjaotused asendada tihedustega.

Töö on jagatud kolmeks peatükiks. Esimeses peatükis anname üldise ülevaate diskreetse ajaga Markovi ahelatest ning defineerime varjatud Markovi mudeli. Kolmandas alapeatükis kirjeldame *forward* ja *backward* muutujaid.

Teises peatükis uurime kahte algoritmi, mille abil saab teadaolevate parameetrite korral leida optimaalse varjatud Markovi ahela realisatsiooni. Optimaalsust kirjeldame kahe erineva kaofunktsiooni kaudu. Esimeses alapeatükis defineerime sümmeetrilise kaofunktsiooni ning kirjeldame selle kaofunktsiooni kaudu leitud riski minimeerivat Viterbi algoritmi. Teises alapeatükis defineerime punktiivilise kaofunktsiooni ning kirjeldame PMAP joondust, mis minimeerib vastava riski.

Kolmandas peatükis uurime parameetrite hindamise iteratiivseid meetodeid. Esimeses alapeatükis kirjeldame Viterbi algoritmile tuginevat meetodit – Viterbi treeningut. Teises alapeatükis kirjeldame meetodit suurima tõepära leidmiseks, mida nimetatakse EM algoritmiks. Kolmandas alapeatükis vaatleme modifikatsiooni Viterbi treeningust, kus iga samm tugineb Viterbi algoritmi asemel PMAP algoritmile. Samuti võrdleme omavahel kolme hindamismeetodit.

Simulatsioonideks kasutatud R-i koodid on toodud töö lisades.

# 1 Varjatud Markovi mudel

## 1.1 Diskreetse ajaga Markovi ahelad

Järgnev alapeatükk tugineb K. Pärna õpikule [3].

Vaatleme ülimalt loenduvat hulka  $S = \{s_1, s_2, \dots\}$  ning juhuslike suuruste jada  $X_1, X_2, X_3, \dots$ , mis võtab väärtusi hulgast  $S$ . Hulka  $S$  kutsutakse olekute ehk seisundite jadaks ning selle elemente olekuteks ehk seisunditeks. Mugavuse mõttes võime  $s_j$  asemel kirjutada lihtsalt  $j$ .

Juhuslike suuruste jada  $\{X_n\}$ ,  $n = 1, 2, 3, \dots$  nimetatakse **Markovi ahelaks**, kui iga  $j, k_1, k_2, \dots, k_{n-1} \in S$  korral

$$P(X_n = j | X_1 = k_1, \dots, X_{n-1} = k_{n-1}) = P(X_n = j | X_{n-1} = k_{n-1}). \quad (1.1)$$

Tingimust (1.1) nimetatakse Markovi omaduseks.

Et  $n$  võtab siin väärtusi  $1, 2, 3, \dots$ , siis on tegemist diskreetse ajaga Markovi ahelatega.

Seisundist  $i$  seisundisse  $j$  ülemineku tõenäosuseks sammul  $n$  nimetatakse tinglikku tõenäosust

$$P(X_n = j | X_{n-1} = k_{n-1}) =: p_{ij}^{(n)}.$$

Markovi ahela algseisund  $X_1$  näitab, millise tõenäosusega üks või teine seisund on ahela algseisundiks ning see antakse ette tõenäosusjaotusega

$$P(X_1 = j) = p_j^1, \quad j = 1, 2, \dots, \quad \sum_j p_j^1 = 1.$$

Markovi ahela üleminekumaatriksiks nimetatakse maatriksit

$$P(n) = (p_{ij}^{(n)}), \quad i, j = 1, 2, 3, \dots$$

Markovi ahelat nimetatakse **homogeenseks**, kui üleminekutõenäosused ei sõltu  $n$  väärtusest, st

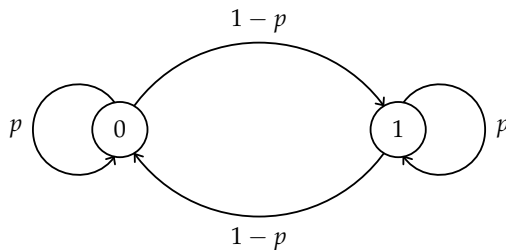
$$p_{ij}^{(n)} =: p_{ij} \quad \forall n$$

Üleminekumaatriksi iga rea summa on alati 1, sest selle maatriksi  $i$ -s rida on seisundite  $j$  tinglik jaotus järgmisel sammul, kui on teada, et praegune seisund on  $i$ .

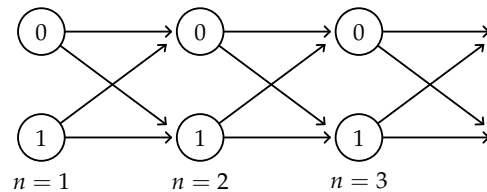
**Näide 1.1.** (Kommunikatsioonimudel). Vaatleme digitaalse info (s.o nullidest ja ühtedest koosnev jada) mitmeetapilist ülekannet. Ülekanne toimub paljude etap-pide kaupa, kus igal sammul võib tekkida viga  $0 \rightarrow 1$  või  $1 \rightarrow 0$  tõenäosusega  $1-p$ .

Oletame, et igal astmel sümbol säilib tõenäosusega  $p$  ja muundub tõenäosusega  $1 - p$ . Sellisel juhul saame sümmeetrilise üleminekumaatriksi

$$P = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}.$$



Joonis 1: Kommunikatsiooni-  
mudeli seisundid ja ülemineku-  
maatriks



Joonis 2: Kommunikatsioonimudeli  
võimalikud realisatsioonid

Joonis 1 kujutab graafiliselt ahela seisundeid ning üleminekumaatriksit. Joonisel 2 näeme Markovi ahela võimalikke realisatsioone.  $n = 3$  korral on erinevaid kombinatsioone kokku  $2^3 = 8$ .

## 1.2 Varjatud Markovi mudel

Varjatud Markovi mudeli definitsiooni võib leida näiteks [4]. Vaatleme homogeen-set Markovi ahelat  $\{X_n\}, n = 1, 2, \dots$ , mis võtab väärtusi lõplikust  $N$ -elemendilisest olekute jadast  $S = \{s_1, s_2, \dots, s_N\}$ . Üleminekumaatriksi tähistame  $A = (a_{ij})$ , kus

$$a_{ij} := P(X_n = j | X_{n-1} = i), \quad n > 1, \quad i, j \in S$$

$$a_{ij} \geq 0, \quad \sum_{j=1}^N a_{ij} = 1.$$

Algtõenäosused  $\pi_j$  tähistame

$$\pi_j := P(X_1 = j).$$

Vaatleme juhuslikku protsessi  $\{Y_n\}, n = 1, 2, \dots$ , mis võtab väärtusi lõplikust  $M$ -elemendilisest hulgast  $O = \{o_1, o_2, \dots, o_M\}$ . Hulga  $O$  elemente nimetame vaatlusteks.

Olgu juhuslik protsess  $(X, Y) = \{(X_n, Y_n)\}_{n \in \mathbb{N}}$  selline, et

- 1) antud  $\{X_n\}$  korral  $\{Y_n\}$  on tinglikult sõltumatud, st iga  $n$  korral

$$\begin{aligned} & P(Y_1 = y_1, \dots, Y_n = y_n | X_1 = i_1, \dots, X_n = i_n) \\ &= \prod_{t=1}^n P(Y_t = y_t | X_1 = i_1, \dots, X_n = i_n), \end{aligned}$$

- 2)  $Y_n$  jaotus sõltub  $\{X_n\}$ -st ainult läbi juhusliku suuruse  $X_n$ , st

$$\prod_{t=1}^n P(Y_t = y_t | X_1 = i_1, \dots, X_n = i_n) = \prod_{t=1}^n P(Y_t = y_t | X_t = i_t).$$

Tingimuste 1) ja 2) kehtides nimetatakse protsessi  $Y$  varjatud Markovi protsessiks ning paari  $(X, Y)$  varjatud Markovi mudeliks (lühidalt HMM, ingl. *Hidden Markov model*). Varjatud Markovi protsess  $Y = \{Y_1, Y_2, \dots\}$  modelleerib vaatlusi, seega juhuslikud suurused  $Y_1, \dots, Y_T$  on selle protsessi esimesed  $T$  elementi. Vaatlustele vastavad  $X_1, \dots, X_T$  realisatsioonid ei ole vaadeldavad, st Markovi ahel on varjatud. Juhuslike suuruste  $X_1, \dots, X_T$  realisatsiooni  $(x_1, \dots, x_T)$  nimetatakse **joonduseks**.

Tingimus 2) kehtib iga  $n$  korral, seega  $Y_n$  jaotus ei sõltu  $n$  väärtusest ega suurusest  $X_s$ , kus  $s \neq n$ . Juhuslikud protsessid  $\{X_n\}$  ja  $\{Y_n\}$  on iga fikseeritud  $n$  korral seotud tinglike tõenäosustega

$$b_j(k) := P(Y_n = o_k | X_n = j).$$

Tõenäosusi  $b_j(k)$  nimetatakse **emissioonitõenäosusteks** ning maatriksit  $B = (b_j(k))$ ,  $j = 1, \dots, N$ ,  $k = 1, \dots, M$  nimetatakse emissioonitõenäosuste maatriksiks. Vastavalt definitsioonile

$$b_j(k) \geq 0, \quad \sum_{k=1}^M b_j(k) = 1.$$

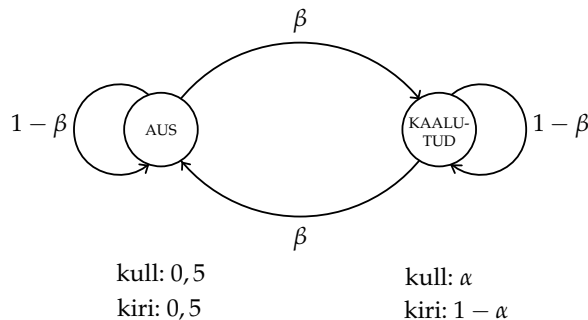
Tingimuse 2) parema poole saame kirja panna kui vastavate emissioonitõenäosuste korrutise:

$$\prod_{t=1}^n P(Y_t = y_t | X_t = i_t) = \prod_{t=1}^n b_{i_t}(y_t).$$

**Näide 1.2.** (Mündivisked). Vaatleme kulli ja kirja viskamist ausa mündiga (kulli saamise tõenäosus on 0,5) ning kaalutud mündiga (kulli saamise tõenäosus on  $\alpha$ ). Mängur vahetab münti tõenäosusega  $\beta$ . Mängija näeb mündiviske tulemust, kuid ei tea, kumma mündiga see tulemus saadi, st olekute (kas aus või kaalutud münt) jada on tundmata. Ülemineku- ja emissioonitõenäosuste maatriksid on vastavalt

$$A = \begin{pmatrix} 1-\beta & \beta \\ \beta & 1-\beta \end{pmatrix}, B = \begin{pmatrix} 0,5 & 0,5 \\ \alpha & 1-\alpha \end{pmatrix}$$

Joonisel 3 esitame ülemineku- ja emissioonitõenäosused ka graafiliselt.



Joonis 3: Mündivisked

Edaspidi tähistame juhuslike suuruste vektorid ning nendele vastavad realisatsioonid järgnevalt:

$$Y_t^T := (Y_t, \dots, Y_T), \quad y_t^T := (y_t, \dots, y_T),$$

$$X_t^T := (X_t, \dots, X_T), \quad x_t^T := (x_t, \dots, x_T).$$

Kui  $t = 1$ , siis tähistame

$$Y^T := (Y_1, \dots, Y_T), \quad y^T := (y_1, \dots, y_T),$$

$$X^T := (X_1, \dots, X_T), \quad x^T := (x_1, \dots, x_T).$$

Varjatud Markovi mudeli rakendamiseks tuleb Rabiner' [2] sõnul kõigepealt lahendada kolm põhiprobleemi:

1. Kuidas antud vaatluste  $y^T$  ning mudeli  $\lambda = (A, B, \pi)$  korral arvutada efektiivselt vaatluste tõenäosus  $P(Y^T = y^T | \lambda)$ ?
2. Kuidas antud vaatluste jada  $y^T$  ning mudeli  $\lambda$  korral valida mingil moel optimaalne joondus  $x^T$ ?
3. Kuidas leida parameetrid  $\lambda = (A, B, \pi)$ , mis maksimeerivad  $P(Y^T = y^T | \lambda)$ ? Need parameetrid on suurima tõepära hinnangud.

Järgnevalt defineerime muutujad, mida nimetatakse *forward* ja *backward* muutujateks, mis lahendavad esimese põhiprobleemi ning aitavad lahendada ka teist ja kolmandat.



### 1.3 *Forward ja backward* muutujad

Olgu meil antud vaatluste jada  $y^T$  ja mudel  $\lambda$ . Soovime arvutada tõenäosust  $P(Y^T = y^T | \lambda)$ . Defineerime suuruse  $\alpha_t(i)$  kui tõenäosuse, et antud mudeli  $\lambda$  korral emiteeritakse osaline vaatluste jada kuni hetkeni  $t$  ning hetkel  $t$  on ahel olekus  $i$ :

$$\alpha_t(i) = P(Y_1 = y_1, \dots, Y_t = y_t, X_t = i | \lambda).$$

Hetkel  $t = 1$  saame

$$\begin{aligned} \alpha_1(i) &= P(Y_1 = y_1, X_1 = i | \lambda) \\ &= P(X_1 = i | \lambda) P(Y_1 = y_1 | X_1 = i, \lambda) = \pi_i b_i(y_1), \quad 1 \leq i \leq N. \end{aligned}$$

Näitame, et saame muutuja  $\alpha_t(i)$  järgnevad väärtused leida rekursiivselt [1].

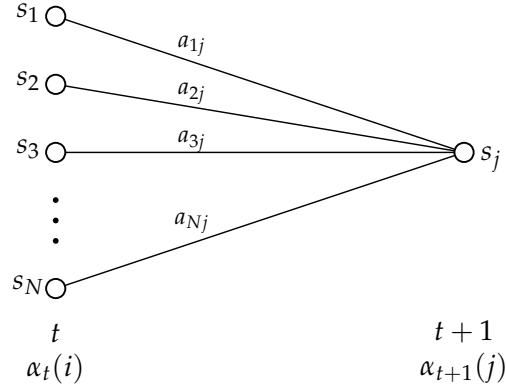
$$\begin{aligned} \alpha_{t+1}(j) &= P(Y_1 = y_1, \dots, Y_{t+1} = y_{t+1}, X_{t+1} = j | \lambda) \\ &= P(Y^{t+1} = y^{t+1}, X_{t+1} = j | \lambda) \\ &= \sum_{i=1}^N P(Y^{t+1} = y^{t+1}, X_t = i, X_{t+1} = j | \lambda) \\ &= \sum_{i=1}^N P(Y^t = y^t, X_t = i | \lambda) \cdot P(Y_{t+1} = y_{t+1}, X_{t+1} = j | Y^t = y^t, X_t = i, \lambda) \\ &= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(X_t = i | \lambda) \cdot P(X_{t+1} = j | Y^t = y^t, X_t = i, \lambda) \cdot \\ &\quad P(Y_{t+1} = y_{t+1} | X_{t+1} = j, Y^t = y^t, X_t = i, \lambda) \\ &= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(X_t = i | \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \\ &\quad \sum_{x^{t-1}} P(X_{t+1} = j, X^{t-1} = x^{t-1} | Y^t = y^t, X_t = i, \lambda) \\ &= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(X_t = i | \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \\ &\quad \sum_{x^{t-1}} P(X_{t+1} = j | X^{t-1} = x^{t-1}, Y^t = y^t, X_t = i, \lambda) \cdot \\ &\quad P(X^{t-1} = x^{t-1} | Y^t = y^t, X_t = i, \lambda) \\ &= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(X_t = i | \lambda) \cdot \\ &\quad \sum_{x^{t-1}} \frac{P(X^{t-1} = x^{t-1}, X_t = i, X_{t+1} = j, Y^t = y^t | \lambda)}{P(X^{t-1} = x^{t-1}, X_t = i, Y^t = y^t | \lambda)} \cdot \\ &\quad P(X^{t-1} = x^{t-1} | Y^t = y^t, X_t = i, \lambda) \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(X_t = i | \lambda) \cdot \\
&\quad \sum_{x^{t-1}} \frac{P(X^{t-1} = x^{t-1}, X_t = i, X_{t+1} = j | \lambda) \cdot P(Y^t = y^t | X^{t-1} = x^{t-1}, X_t = i, \lambda)}{P(X^{t-1} = x^{t-1}, X_t = i | \lambda) \cdot P(Y^t = y^t | X^{t-1} = x^{t-1}, X_t = i, \lambda)} \cdot \\
&\quad P(X^{t-1} = x^{t-1} | Y^t = y^t, X_t = i, \lambda) \\
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(X_t = i | \lambda) \cdot \\
&\quad \sum_{x^{t-1}} \frac{P(X_{t+1} = j | X^{t-1} = x^{t-1}, X_t = i, \lambda) \cdot P(X^{t-1} = x^{t-1}, X_t = i | \lambda)}{P(X^{t-1} = x^{t-1}, X_t = i | \lambda)} \cdot \\
&\quad P(X^{t-1} = x^{t-1} | Y^t = y^t, X_t = i, \lambda) \\
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(X_t = i | \lambda) \cdot \\
&\quad P(X_{t+1} = j | X_t = i, \lambda) \\
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot \\
&\quad P(X_t = i, X_{t+1} = j | \lambda) \\
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(X_t = i, X_{t+1} = j | \lambda) \cdot b_j(y_{t+1}) \\
&= \sum_{i=1}^N P(Y^t = y^t | X_t = i, \lambda) \cdot P(X_t = i | \lambda) \cdot P(X_{t+1} = j | X_t = i, \lambda) \cdot b_j(y_{t+1}) \\
&= \sum_{i=1}^N P(Y^t = y^t, X_t = i, \lambda) \cdot P(X_{t+1} = j | X_t = i, \lambda) \cdot b_j(y_{t+1}) \\
&= \sum_{i=1}^N \alpha_t(i) a_{ij} b_j(y_{t+1})
\end{aligned}$$

Kokkuvõttes

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(y_{t+1}), \quad 1 \leq t \leq T-1 \\
1 \leq j \leq N.$$

Suurust  $\alpha_t(i)$  nimetatakse *forward* muutujaks.



Joonis 4: *Forward* muutuja leidmine rekursiivselt ( joonis tehtud [2] põhjal)

Joonis 4 illustreerib suuruse  $\alpha_{t+1}(j)$  leidmiseks vajalikke samme. Hetkel  $t+1$  võib olekusse  $s_j$  jõuda  $N$  erinevast olekust hetkel  $t$ . Suurus  $\alpha_t(i)a_{ij}$  kirjeldab ühistõenäosust, et nägime vaatlusi  $y_1, \dots, y_t$  ning hetkel  $t+1$  jõudsimise olekusse  $s_j$  olekust  $s_i$ . Seega  $\alpha_{t+1}(j)$  leidmiseks peame suurust  $\alpha_t(i)a_{ij}$  summeerima üle kõigi  $N$  oleku  $s_i$  ning korrutama saadud tulemuse üleminekutõenäosusega  $b_j(y_{t+1})$ .

Kuna definitsiooni kohaselt

$$\alpha_T(i) = P(Y^T = y^T, X_T = i | \lambda),$$

siis otsitav  $P(Y^T = y^T | \lambda)$  avaldub kujul

$$P(Y^T = y^T | \lambda) = \sum_{i=1}^N \alpha_T(i).$$

Analoogiliselt defineerime *backward* muutuja  $\beta_t(i)$ , kui tõenäosuse, et mudeli  $\lambda$  ning hetkel  $t$  oleku  $i$  korral näeme osalist vaatluste jada hetkest  $t+1$  kuni lõpphetkeni  $T$ :

$$\beta_t(i) = P(Y_{t+1} = y_{t+1}, \dots, Y_T = y_T | X_t = i, \lambda).$$

Hetkel  $t = T$  väärtustame defineeritud suuruse ühega:

$$\beta_T(i) = 1.$$

Tuletame suuruse  $\beta_t(i)$  arvutamise valemi iga  $t = T-1, \dots, 1$  korral [1].

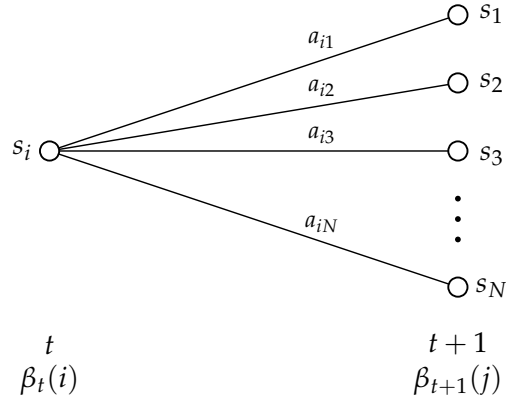
$$\begin{aligned} \beta_t(i) &= P(Y_{t+1} = y_{t+1}, \dots, Y_T = y_T | X_t = i, \lambda) \\ &= P(Y_{t+1}^T = y_{t+1}^T | X_t = i, \lambda) \\ &= \frac{P(Y_{t+1}^T = y_{t+1}^T, X_t = i | \lambda)}{P(X_t = i | \lambda)} \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^N \frac{P(Y_{t+1}^T = y_{t+1}^T, X_t = i, X_{t+1} = j | \lambda)}{P(X_t = i | \lambda)} \\
&= \sum_{j=1}^N P(Y_{t+1}^T = y_{t+1}^T | X_t = i, X_{t+1} = j, \lambda) \cdot \frac{P(X_t = i, X_{t+1} = j | \lambda)}{P(X_t = i | \lambda)} \\
&= \sum_{j=1}^N P(Y_{t+1}^T = y_{t+1}^T | X_t = i, X_{t+1} = j, \lambda) \cdot P(X_{t+1} = j | X_t = i, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+1}^T = y_{t+1}^T | X_{t+1} = j, \lambda) \cdot P(X_{t+1} = j | X_t = i, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+1} = y_{t+1}, Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \\
&= \sum_{j=1}^N P(Y_{t+1} = y_{t+1} | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, \lambda) \cdot \\
&\quad P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \\
&= \sum_{j=1}^N P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \cdot \\
&\quad \sum_{x_{t+2}^T} P(Y_{t+1} = y_{t+1}, X_{t+2}^T = x_{t+2}^T | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \cdot \\
&\quad \sum_{x_{t+2}^T} P(Y_{t+1} = y_{t+1} | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, X_{t+2}^T = x_{t+2}^T, \lambda) \cdot \\
&\quad P(X_{t+2}^T = x_{t+2}^T | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \cdot \\
&\quad \sum_{x_{t+2}^T} \frac{P(Y_{t+1}^T = y_{t+1}^T | X_{t+1} = j, X_{t+2}^T = x_{t+2}^T, \lambda)}{P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, X_{t+2}^T = x_{t+2}^T, \lambda)} \cdot \\
&\quad P(X_{t+2}^T = x_{t+2}^T | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij} \cdot \\
&\quad \sum_{x_{t+2}^T} \frac{P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(Y_{t+2}^T = y_{t+2}^T | X_{t+2}^T = x_{t+2}^T, \lambda)}{P(Y_{t+2}^T = y_{t+2}^T | X_{t+2}^T = x_{t+2}^T, \lambda)} \cdot \\
&\quad P(X_{t+2}^T = x_{t+2}^T | Y_{t+2}^T = y_{t+2}^T, X_{t+1} = j, \lambda) \\
&= \sum_{j=1}^N P(Y_{t+1} = y_{t+1} | X_{t+1} = j, \lambda) \cdot P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot a_{ij}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^N P(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = j, \lambda) \cdot b_j(y_{t+1}) \cdot a_{ij} \\
&= \sum_{j=1}^N a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)
\end{aligned}$$

Kokkuvõttes

$$\begin{aligned}
\beta_t(i) &= \sum_{j=1}^N a_{ij} b_j(y_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1 \\
&1 \leq i \leq N.
\end{aligned}$$



Joonis 5: *Backward* muutuja leidmine rekursiivselt (joonis tehtud [2] põhjal)

Nagu jooniselt 5 näha, siis selleks, et hetkel  $t$  olla olekus  $s_i$ , peame läbi vaatama kõik  $N$  võimalikku olekut hetkel  $t+1$ , võttes arvesse üleminekutõenäosused  $a_{ij}$  ning emissioonitõenäosused  $b_j(y_{t+1})$ . Suuruse  $\beta_t(i)$  leidmiseks peame seega summeerima korrutist  $a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)$  üle kõikide olekute  $s_j$ .

*Forward* ja *backward* muutujate arvutamiseks tuleb meil igal sammul korrutada kokku  $T-1$  üleminekutõenäosust ning  $T$  emissioonitõenäosust. Kuna kõik elemendid on ühest väiksemad, siis  $T$  kasvades  $\alpha_t(i)$  ja  $\beta_t(i)$  väärtused lähenevad eksponentsiaalselt nullile ning arvutid ümardavad vastavad suurused nulliks. Probleemi lahendamiseks kasutatakse skaleerimist. Skaleerimiseks korrutatakse vastav suurus ainult indeksist  $t$  sõltuva koefitsendiga. Defineerime skaleeritud muutujad iga  $1 \leq j \leq N$  korral järgnevalt [1]:

$$\begin{aligned}
\tilde{\alpha}_t(j) &= \frac{P(Y_1 = y_1, \dots, Y_t = y_t, X_t = j)}{P(Y_1 = y_1, \dots, Y_t = y_t)} \\
\tilde{\beta}_t(j) &= \frac{P(Y_{t+1} = y_{t+1}, \dots, Y_T = y_T | X_t = j)}{P(Y_{t+1} = y_{t+1}, \dots, Y_T = y_T | Y_1 = y_1, \dots, Y_t = y_t)}
\end{aligned}$$

Muutujate väärtused saame leida rekursiivselt [1]:

$$\begin{aligned}
1. \quad & \tilde{\alpha}_1(j) = \frac{\pi_j b_j(y_1)}{\sum_{i=1}^N \pi_i b_i(y_1)} \\
& \tilde{\beta}_T(i) = 1 \\
2. \quad & \tilde{\alpha}_t(j) = c_t \left[ \sum_{i=1}^N \tilde{\alpha}_{t-1}(i) a_{ij} \right] b_j(y_t), \quad 2 \leq t \leq T \\
& \tilde{\beta}_t(i) = c_t \sum_{j=1}^N a_{ij} b_j(y_{t+1}) \tilde{\beta}_{t+1}(j), \quad 1 \leq t \leq T-1,
\end{aligned}$$

kus skaleerimiskoeffitsent on

$$c_t = \frac{1}{\sum_{j=1}^N \sum_{i=1}^N \tilde{\alpha}_{t-1}(i) a_{ij} b_j(y_t)}.$$

## 2 Segmenteerimine

Olgu  $y^T$  antud vaatlused. Sellele vastav joondus ei ole teada, st Markovi ahel on varjatud, aga eeldame, et teame täpselt HMM parameetrite  $\lambda = (A, B, \pi)$  väärtusi. Segmenteerimiseks nimetatakse varjatud joonduse  $(x_1, \dots, x_T)$  hinnangu leidmist. Teisisõnu, otsime kujutust  $g : S^T \rightarrow O^T$ , mis seab igale vaatluste jadale  $y^T$  vastavusse joonduse  $g(y^T)$ . Kuna Markovi ahela  $X^T$  realisatsiooni on võimatu leida täpselt, peab joondus  $g(y^T)$  olema mingil moel optimaalne. Optimaalsuskriteeriumi võib sõnastada mitmeti. Kirjeldame konspektis [4] välja toodud ideed. Eesmärk võib olla leida joondus, millel on suurim tõenäosus või mille klassifitseerimisvigade arv oleks minimaalne. Et hinnata  $g$  kvaliteeti, hindame kõigepealt iga võimaliku üksiku joonduse  $z^T = (z_1, \dots, z_T) \in S^T$  headust läbi riskifunktsiooni. Antud  $y^T$  korral defineerime joonduse  $z^T$  riski kui funktsiooni, mis seab joondusele  $z^T$  vastavusse ühe reaalarvu, tähistame selle riski  $R(z^T|y^T)$ :

$$R(z^T|y^T) : S^T \rightarrow [0, \infty).$$

Segmenteerimisprobleemi lahendamiseks tuleb meil leida mingil moel optimaalne joondus, seega otsime joondust, mille risk oleks minimaalne:

$$g(y^T) := \operatorname{argmin}_{z^T \in S^T} R(z^T|y^T).$$

Üks võimalus on riski defineerida kaofunktsiooni  $L$  kaudu:

$$L : S^T \times S^T \rightarrow [0, \infty),$$

kus  $L(a^T, b^T)$  näitab kadu, mis tekkis hinnates joonduseks  $a^T$ , kui tegelik joondus oli  $b^T$ . Iga joonduse  $z^T \in S^T$  korral risk  $R(z^T|y^T)$  on kaofunktsiooni  $L$  tinglik keskvääratus, teades, et  $Y^T = y^T$ :

$$R(z^T|y^T) := E[L(z^T, X^T)|Y^T = y^T] = \sum_{x^T \in S^T} L(z^T, x^T)P(X^T = x^T|Y^T = y^T).$$

Järgnevalt vaatame kahte enamlevinud kaofunktsiooni.

## 2.1 Viterbi algoritm

Defineerime sümmeetrilise kaofunktsiooni järgnevalt:

$$L(z^T, x^T) = \begin{cases} 0, & \text{kui } z^T = x^T; \\ 1, & \text{kui } z^T \neq x^T. \end{cases} \quad (2.1)$$

Kaofunktsioon (2.1) saab väärtuse 0, kui joondused  $x^T$  ja  $z^T$  ühtivad ning väärtuse 1, kui joondused erinevad ühe või mitme elemendi poolest. Seega võime kaofunktsiooni indikaatorfunktsiooni kaudu kirja panna kui  $L(z^T, x^T) = I_{\{x^T \neq z^T\}}$  ning tinglik risk avaldub kujul

$$\begin{aligned} R(z^T|y^T) &= E[I_{\{X^T \neq z^T\}}|Y^T = y^T] = P(X^T \neq z^T|Y^T = y^T) \\ &= 1 - P(X^T = z^T|Y^T = y^T). \end{aligned} \quad (2.2)$$

Defineerime funktsiooni  $v$ , mis minimeerib riski (2.2) ning leiab suurima tõenäosusega joonduse:

$$v(y^T) := \operatorname{argmax}_{z^T \in S^T} P(X^T = z^T|Y^T = y^T). \quad (2.3)$$

Joondust  $v(y^T)$  nimetatakse Viterbi joonduseks (dünaamilise planeerimise algoritmi – Viterbi algoritmi järgi). On ilmne, et Viterbi joondus ei pruugi olla ühene. Seega Viterbi joonduseks nimetame iga joondust  $v^T = (v_1, \dots, v_T)$ ,  $v_i \in S$ , mis maksimeerib tingliku tõenäosuse (2.3).

Kirjeldame algoritmi, mille abil saab leida suurima tingliku tõenäosusega joondust (2.3), mida nimetatakse Viterbi algoritmiks.

Tänu võrdusele

$$P(X^T = z^T|Y^T = y^T) = \frac{P(X^T = z^T, Y^T = y^T)}{P(Y^T = y^T)}$$

võime joonduse (2.3) leidmisel tinglike tõenäosuse maksimeerimise asemel maksimeerida ühistõenäosust:

$$\operatorname{argmax}_{z^T \in S^T} P(X^T = z^T|Y^T = y^T) = \operatorname{argmax}_{z^T \in S^T} P(X^T = z^T, Y^T = y^T).$$

Otsitava joonduse  $x^T = (x_1, \dots, x_T)$  leidmiseks defineerime iga  $t = 1, \dots, T$  ja iga oleku  $i \in S$  korral suuruse  $\delta_t(i)$  järgmiselt:

$$\delta_t(i) = \max_{x_1, \dots, x_{t-1}} P(Y_1 = y_1, \dots, Y_t = y_t, X_1 = x_1, \dots, X_{t-1} = x_{t-1}, X_t = i).$$

Algväärtus  $\delta_1(i)$  näitab iga seisundi kohta tõenäosust olla alghetkel olekus  $i$  ning emiteerida vaatlus  $y_1$ . Seega

$$\delta_1(i) = P(X_1 = i, Y_1 = y_1) = \pi_i b_i(y_1), \quad 1 \leq i \leq N.$$

Suuruse  $\delta_t(i)$  järgnevad väärtused saame leida induktsiooni kaudu [1].

**Lause 1.** Iga  $2 \leq t \leq T$  korral

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] b_j(y_{t+1}).$$

*Tõestus.* Kasutades  $\{X_t\}$  ja  $\{Y_t\}$  tinglikku sõltumatust ning Markovi omadust saame

$$\begin{aligned} \delta_{t+1}(j) &= \max_{x_1, \dots, x_t} P(Y_1 = y_1, \dots, Y_{t+1} = y_{t+1}, X_1 = x_1, \dots, X_t = x_t, X_{t+1} = j) \\ &= \max_{x_1, \dots, x_t} P(Y^{t+1} = y^{t+1}, X^t = x^t, X_{t+1} = j) \\ &= \max_{x_1, \dots, x_t} P(Y^{t+1} = y^{t+1} | X^t = x^t, X_{t+1} = j) \cdot P(X^t = x^t, X_{t+1} = j) \\ &= \max_{x_1, \dots, x_t} \prod_{k=1}^t P(Y_k = y_k | X_k = x_k) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j) \cdot \\ &\quad P(X_{t+1} = j | X^t = x^t) \cdot P(X^t = x^t) \\ &= \max_{x_1, \dots, x_t} \prod_{k=1}^t P(Y_k = y_k | X_k = x_k) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j) \cdot \\ &\quad P(X_{t+1} = j | X_t = x_t) \cdot P(X^t = x^t) \\ &= \max_{x_1, \dots, x_t} \prod_{k=1}^t P(Y_k = y_k | X_k = x_k) \cdot P(X^t = x^t) \cdot \\ &\quad P(X_{t+1} = j | X_t = x_t) \cdot P(Y_{t+1} = y_{t+1} | X_{t+1} = j) \\ &= \max_{x_1, \dots, x_t} P(Y^t = y^t, X^t = x^t) \cdot a_{x_t j} \cdot b_j(y_{t+1}) \\ &= \max_{x_t} \left( \max_{x_1, \dots, x_{t-1}} P(Y^t = y^t, X^{t-1} = x^{t-1}, X_t = x_t) \cdot a_{x_t j} \cdot b_j(y_{t+1}) \right) \\ &= \max_{1 \leq i \leq N} \left( \max_{x_1, \dots, x_{t-1}} P(Y^t = y^t, X^{t-1} = x^{t-1}, X_t = i) \cdot a_{ij} \cdot b_j(y_{t+1}) \right) \end{aligned}$$

Seega

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] b_j(y_{t+1}), \quad 2 \leq t \leq T.$$

□



Otsitava joonduseni jõudmiseks peame iga  $t$  ja  $j$  korral jätma meelde argumendi, mis realiseeris suuruse  $\delta_t(j)$ . Selleks defineerime muutuja  $\psi_t(j)$ , mis jätab meelde indeksi  $i$ , mis hetkel  $t$  maksimeeris parima olekute jada tõenäosuse:

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} \delta_t(i) a_{ij}.$$

Lõppväärtuseks saame

$$v_T = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)],$$

kus  $v_T$  näitab optimaalse joonduse viimast seisundit. Tagant poolt ette poole liikudes leiame joonduse eelviimase, eel-eelviimase, kuni esimese elemendi:

$$v_t = \psi_t(v_{t+1}), \quad t = T-1, T-2, \dots, 1.$$

Algoritm väljastab joonduse  $v^T$ , mida antud mudeli ja vaatluste jada korral näeme kõige tõenäolisemalt. Kuna Viterbi joondus ei pruugi olla ühene, siis valib algoritm alati leksikograafiliselt väiksema oleku, kui mitme joonduse esinemise tõenäosused on võrdsed. See tagab, et samade sisendandmete korral ei väljasta Viterbi algoritm kunagi erinevaid joondusi.

1. **Sisend:**  $\lambda = (A, B, \pi)$ ,  $y^T = (y_1, \dots, y_T)$
2. Iga  $i \in S$  korral defineeri  $\delta_1(i) := \pi_i b_i(y_1)$
3. Iga  $t = 1, \dots, T-1$  korral
  - $\delta_{t+1}(j) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] b_j(y_{t+1})$
  - $\psi_t(j) := \operatorname{argmax}_{1 \leq i \leq N} [\delta_t(i) a_{ij}]$
4. **Väljund:** Leia Viterbi joondus  $v^T$  tagant ette poole liikudes, valides igal sammul leksikograafiliselt vähim sümbol:
  - $v_T := \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$
  - $v_t = \psi_t(v_{t+1})$ ,  $t = T-1, \dots, 1$

Algoritm 1: Viterbi algoritm

**Näide 2.1.** Vaatleme kulli ja kirja viskamist kolme erineva mündiga. Vastava HMM parameetrid on järgmised:

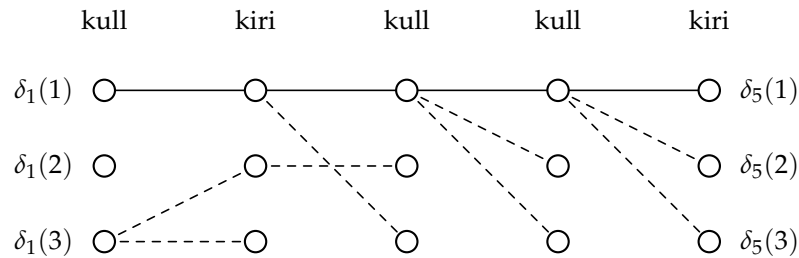
$$A = \begin{pmatrix} 0,6 & 0,2 & 0,2 \\ 0,2 & 0,5 & 0,3 \\ 0,3 & 0,3 & 0,4 \end{pmatrix}, B = \begin{pmatrix} 0,5 & 0,5 \\ 0,3 & 0,7 \\ 0,6 & 0,4 \end{pmatrix}, \pi = (0,4; 0,3; 0,3).$$

Münti visati 5 korda ning saadi vaatlused  $y^5 = (\text{kull}, \text{kiri}, \text{kull}, \text{kull}, \text{kiri})$ . Leiame Viterbi algoritmi abil, millist münti kasutati igal viskel kõige tõenäolisemalt. Arvutame vastavad  $\delta_t(i)$  ja  $\psi_t(i)$  väärtused ning kirjutame need allolevasse tabelisse.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$\delta_t(1)$	0,2	0,06	0,018	0,0054	0,00162
$\delta_t(2)$	0,09	0,0378	0,00567	0,00108	0,000756
$\delta_t(3)$	0,18	0,0288	0,0072	0,00216	0,000432
$\psi_t(1)$	1	1	1	1	—
$\psi_t(2)$	3	2	1	1	—
$\psi_t(3)$	3	1	1	1	—

Tabel 1:  $\delta_t(i)$  ja  $\psi_t(i)$  väärtused

Tabelist 1 näeme, et Viterbi joonduseks saame  $v^5 = (1, 1, 1, 1, 1)$ , mis tähendab, et kõige suurema tõenäosusega tehti kõik visked ausa mündiga.



Joonis 6: Viterbi algoritmi käik

Joonis 6 illustreerib Viterbi algoritmi käiku eeloleva näite korral. Pidev joon näitab optimaalset joondust ning katkendlik joon kirjeldab suuruse  $\psi_t(j)$  väärtusi.

Viterbi algoritm leiab küll suurima tingliku tõenäosusega joonduse, kuid see ei maksimeeri oodatavat arvu õigeid olekuid. Järgnevas alapeatükis leiame sellise joonduse, mis maksimeeriks oodatava arvu õigeid olekuid.

## 2.2 PMAP joondus

Sümmeetriline kaofunktsioon (2.1) ei erista, kas  $z^T$  ja  $y^T$  erinevad ühe või mitme elemendi poolest. Järgnevalt võrdleme joondusi elementhaaval.

Defineerime kaofunktsiooni järgmiselt:

$$L(z^T, x^T) := \sum_{t=1}^T l(z_t, x_t),$$

kus  $l : S \times S \rightarrow [0, \infty)$  on punktiviisiline kaofunktsioon:

$$l(z, x) = \begin{cases} 0, & \text{kui } z = x; \\ 1, & \text{kui } z \neq x. \end{cases}$$

Kaofunktsioon  $L$  loendab elementhaaval joonduste  $z^T$  ja  $y^T$  erinevusi ning selle saab indikaatorfunktsiooni kaudu kirja panna kui  $L(z^T, x^T) = \sum_{t=1}^T I_{\{x_t \neq z_t\}}$ , seega risk

$$\begin{aligned} R(z^T|y^T) &= E[L(z^T, X^T)|Y^T = y^T] = \sum_{t=1}^T E[l(z_t, X_t)|Y^T = y^T] \\ &= \sum_{t=1}^T E[I_{\{X_t \neq z_t\}}|Y^T = y^T] = \sum_{t=1}^T \left(1 - P(X_t = z_t|Y^T = y^T)\right) \\ &= T - \sum_{t=1}^T P(X_t = z_t|Y^T = y^T) \end{aligned} \quad (2.4)$$

mõõdab  $z^T$  klassifitseerimisvigade oodatavat arvu antud  $y^T$  korral.

Joondus

$$u_t(y^T) := \operatorname{argmax}_{z \in S} P(X_t = z|Y^T = y^T), t = 1, \dots, T \quad (2.5)$$

minimeerib riski (2.4) ning on seega joondus, millel on madalaim klassifitseerimisvigade arv. Joondust  $u$  nimetatakse PMAP (ingl. *pointwise maximum a posteriori*) joonduseks.

Kirjeldame algoritmi, mille abil leida PMAP joondus.

Selleks defineerime muutuja  $\gamma_t(i)$  järgnevalt:

$$\gamma_t(i) = P(X_t = i|Y^T = y^T).$$

Tingliku tõenäosuse definitsiooni põhjal

$$P(X_t = i|Y^T = y^T) = \frac{P(Y^T = y^T, X_t = i)}{P(Y^T = y^T)} = \frac{P(Y^T = y^T, X_t = i)}{\sum_{j=1}^N P(Y^T = y^T, X_t = j)}.$$

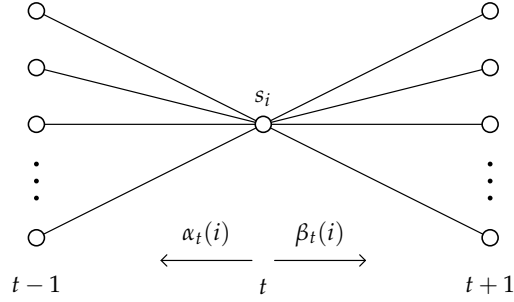
Peatükis 1.3 saadud tulemuste põhjal saame suurst  $\gamma_t(i)$  kirjeldada *forward* ja *backward* valemite kaudu:

$$\alpha_t(i)\beta_t(i) = P(Y^t = y^t, X_t = i) \cdot P(Y_{t+1}^T = y_{t+1}^T | X_t = i) = P(Y^T = y^T, X_t = i).$$

Seega

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)}.$$

Muutuja  $\gamma_t(i)$  leidmist *forward* ja *backward* valemite kaudu illustreerib joonis 7.



Joonis 7: Sammud  $\gamma_t(i)$  leidmiseks

Kasutades abimuutujat  $\gamma_t(i)$  saame leida hetkel  $t$  kõige tõenäolisema oleku  $u_t$  väärtuse:

$$u_t = \operatorname{argmax}_{1 \leq i \leq N} \gamma_t(i), \quad 1 \leq t \leq T.$$

Saadud joondus  $u^T = (u_1, \dots, u_T)$  ei pruugi olla ühene, seega igal sammul, kus mitme oleku esinemise tõenäosused on võrdsed, valime leksikograafiliselt väikseima oleku.

Valides iga  $t$  korral kõige tõenäolisema oleku, maksimeerib (2.5) oodatava arvu õigeid olekuid, kuid saadava joondusega võib esineda teatud probleeme. Kuna PMAP joonduse leidmisel ei arvestata üleminekutõenäosustega, siis juhul kui mingi  $i$  ja  $j$  korral üleminekutõenäosus  $a_{ij} = 0$ , võib saadava joonduse tõenäosus olla väga madal või isegi null.

**Näide 2.2.** Vaatleme täringuviskeid kolme täringuga: ühe ausa ning kahe kaalutud täringuga. Tähistame need täringud vastavalt 1, 2 ja 3. Vastava HMM parameetrid on järgmised:

$$A = \begin{pmatrix} 0,8 & 0,1 & 0,1 \\ 0,0 & 0,6 & 0,4 \\ 0,4 & 0,6 & 0,0 \end{pmatrix}, \quad B = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \\ 1/2 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 \end{pmatrix},$$

$$\pi = (1/3, 1/3, 1/3).$$

Täringuid visatakse 100 korda ning tulemuseks saadakse järgmised vaatlused:

(3, 6, 6, 4, 2, 2, 1, 4, 1, 6, 2, 4, 5, 6, 5, 3, 4, 4, 5, 6, 6, 2, 4, 4, 2, 6, 3, 1, 3, 3, 6, 6, 4, 3, 5, 5, 2, 2, 2, 6, 1, 3, 4, 2, 6, 2, 3, 4, 4, 5, 6, 4, 1, 1, 4, 6, 1, 6, 6, 6, 1, 6, 5, 2, 4, 2, 3, 6, 5, 2, 6, 1, 6, 1, 6, 6, 6, 5, 5, 4, 2, 2, 5, 3, 6, 6, 6, 6, 2, 6, 4, 2, 1, 3, 4, 6, 5, 4, 1, 3)

Võrdleme Viterbi ja PMAP joondust õige joondusega.

Tegelik: (1, 1, 1, 1, 1, 1, 1, 1, 3, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 2, 2, 2, 2, 3, 1, 1, 2, 2, 2, 3, 1, 1, 1, 1, 3, 2, 3, 2, 2, 3, 1, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 1, 1, 1, 1, 1, 2, 3, 2, 2, 2, 2, 3, 1, 1, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 1, 3, 2, 3, 1, 3, 1)

Viterbi: (1,  
1, 3, 2, 2, 3, 2,  
2, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1)

PMPA: (3, 2, 2, 2, 1, 1, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 2,  
2, 2, 3, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 3, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,  
2, 2, 3, 1, 1, 1, 1, 1, 1, 1)

Üleminekumaatriksilt  $A$  näeme, et tõenäosus  $a_{11}$  on suurem kui teised elemendid. Kui emissioonitõenäosused oleksid kõik võrdsed, siis oleks joonduse leidmisel oluline vaid üleminekumaatriks. Sel juhul väljastaks Viterbi algoritm ainult ühtede jada. Emissioonitõenäosuste erinevus ühtlustab joondust. Sellest hoolimata on Viterbi joondusele tihti omane, et see jääb pikalt pidama olekule  $i$ , mille üleminekutõenäosus  $a_{ii}$  on suurem kui teiste olekute korral. Eelolevas näites esinebki olek 1 pikkade blokkidena.

Antud näites leidub PMAP joonduses üleminek olekust 2 olekusse 1, kuigi üleminekumaatriksilt on näha, et selline üleminek ei ole tegelikult võimalik, seega selle joonduse tõenäosus on null.

### 3 Mudeli parameetrite hindamine

Rabiner' kolmest põhiprobleemist on mudeli parameetrite  $\lambda = (A, B, \pi)$  hindamine kõige keerulisem. Probleem seisneb selles, et varjatud andmetega mudelite korral ei pruugi tõepärafunktsioonil olla ilmutatud kuju või on funktsioon keerulisel kujul ning seda on raske maksimeerida. Hinnangute leidmiseks on palju erinevaid meetodeid. Üks levinuim meetod on suurima tõepära meetod, mis maksimeerib  $P(Y^T = y^T | \lambda)$ . Tõepärafunktsioonist võetakse tuletis, see võrdsustatakse nulliga ning leitakse ekstreemumkoht. Kuna varjatud Markovi mudelis on peidetud andmed, siis maksimeeritav tõepärafunktsioon avaldub logaritmide summana üle kõikide joonduste  $x$ , millest on tuletist raske leida. Tõepärafunktsiooni otse maksimeerimise asemel saame parameetrite suurima tõepära hindamiseks kasutada iteratiivseid hindamismeetodeid ehk treeninguid. Iteratiivsetele meetoditele antakse ette parameetrite algväärtused, igal iteratsioonil kohandatakse parameetrite väärtusi eelnevate põhjal. Kõige populaarsem iteratiivne algoritm suurima tõepära leidmiseks varjatud andmetega mudelite puhul on EM algoritm (ingl. *Expectation-Maximization algorithm*). EM algoritmi abil on hinnangute leidmine aga üldiselt üsna keerukas ning alternatiivina saame kasutada Viterbi treeningut, mida on oluliselt lihtsam implementeerida, kuid mis annab sageli üsna äärmuslikud hinnangud.

Mudeli parameetrite hindamise algoritmidele on väga olulised alglähendid. Tõepärafunktsioonil on üldiselt mitu ekstreemumkohta ning kehvade algparameetrite valiku korral ei pruugi me jõuda globaalsesse maksimumi. Et leida head alglähendid ning jõuda globaalsesse maksimumi, tuleks anda ette palju erinevaid alglähendeid ning nende hulgast valida suurima tõepäraga parameetrid. Alglähendid tuleks valida võimalikult erinevad, et need kataksid ühtlaselt kogu otsimispiirkonna.

Kuna käesolevas peatükis on vaatluste  $y^T$  ning joonduse  $x^T$  pikkus  $T$  fikseeritud, siis tähistame

$$y := (y_1, \dots, y_T), \quad x := (x_1, \dots, x_T)$$

ning vastavad tõenäosused

$$P(y) := P(Y_1 = y_1, \dots, Y_T = y_T), \quad P(x) := P(X_1 = x_1, \dots, X_T = x_T).$$

#### 3.1 Viterbi treening

Viterbi treeningu kirjelduse võib leida näiteks [1]. Olgu  $y$  teadaolev vaatluste jada ning meie eesmärk on hinnata mudeli  $\lambda = (A, B, \pi)$  parameetreid. Viterbi treening on iteratiivne algoritm, mis maksimeerib vaatluste ning varjatud joonduse ühistõepära, kuid ei pruugi leida suurima tõepära hinnangut  $P(y | \lambda)$ . Viterbi treening leiab  $l$ -ndal iteratsioonil Viterbi joonduse  $v^{(l)}$  parameetrite  $\lambda^{(l-1)}$  põhjal ning seejärel leitakse parameetrite uued väärtused  $\lambda^{(l)}$  joonduse  $v^{(l)}$  kaudu.

Igal iteratsioonil kohandatakse kordamööda joonduse  $v^{(l)}$  ning parameetrite  $\lambda^{(l)}$  väärtusi. Kuna Viterbi treening ei suuda efektiivselt hinnata algtõenäosusi, siis käesolevas alapeatükis eeldame, et  $\pi$  on teada. Tähistame hinnatavad parameetrid  $\theta := (A, B)$ .

**Sisend:**  $A, B$  algväärtused, algoleku tõenäosused  $\pi$ , vaatluste vektor  $y$

1. Leia Viterbi joondus  $(v_1, \dots, v_T) = v^{(1)}$

2. Viterbi joonduse  $v^{(l)}$  korral leia  $A^{(l)}, B^{(l)}$  järgmise eeskirja alusel:

$$(a) \begin{cases} a_{ij}^{(l)} = \hat{P}(X_l = s_j | X_{l-1} = s_i) = \frac{T_{ij}^{(l)}}{\sum_k T_{ik}^{(l)}}, \text{ kui } \sum_k T_{ik}^{(l)} \neq 0 \\ a_{ij}^{(l)} = a_{ij}^{(l-1)}, \text{ kui } \sum_k T_{ik}^{(l)} = 0 \end{cases}, \text{ kus}$$

$$T_{ij}^{(l)} = \#\{t : v_t^{(l)} = s_i, v_{t+1}^{(l)} = s_j\} = \sum_{t=1}^{T-1} I_{\{(s_i, s_j)\}}(v_t^{(l)}, v_{t+1}^{(l)})$$

$$(b) \begin{cases} b_i(k)^{(l)} = \hat{P}(Y_l = o_k | X_l = s_i) = \frac{E_{ik}^{(l)}}{\sum_j E_{ij}^{(l)}}, \text{ kui } \sum_j E_{ij}^{(l)} \neq 0 \\ b_i(k)^{(l)} = b_i(k)^{(l-1)}, \text{ kui } \sum_j E_{ij}^{(l)} = 0 \end{cases}, \text{ kus}$$

$$E_{ij}^{(l)} = \#\{t : v_t^{(l)} = s_i, y_t = o_j\} = \sum_{t=1}^T I_{\{(s_i, o_j)\}}(v_t^{(l)}, y_t)$$

3.  $A^{(l)}, B^{(l)}$  põhjal leia uus joondus  $v^{(l+1)}$

4. Lõpeta, kui  $v^{(l+1)} = v^{(l)}$

**Väljund:** parameetrite  $A$  ja  $B$  hinnangud

Algoritm 2: Viterbi treening

Esialgselt ette antud parameetrite väärtuste korral leiab Viterbi treening kõigepealt suurima tingliku tõenäosusega Viterbi joonduse  $v^{(1)} = (v_1, \dots, v_T)$ . Üleminekumatriksi elemendi  $a_{ij}$  hindamiseks loeme kokku kõik üleminekud olekust  $i$  olekusse  $j$  ning jagame saadud arvu kõikide üleminekutega olekust  $i$  olekusse  $k \in S$ . Analoogiliselt  $b_i(k)$  leidmiseks loeme kokku, mitu korda emiteeriti sümbol  $o_k$  olekust  $i$  ning jagame tulemuse arvuga, mis näitab, mitu korda emiteeriti sümbol  $o_l \in O$  olekust  $i$ . Nüüd itereerime, kasutades uusi hinnanguid  $\hat{A}^{(l)}, \hat{B}^{(l)}$  leiame uue joonduse  $v^{(l+1)}$  ning hindame mudeli parameetreid  $A^{(\hat{l}+1)}, B^{(\hat{l}+1)}$  sama skeemi alusel. Selgub, et algoritmi iteratsioonide arv on alati lõplik.

**Lause 2.** Viterbi treeningu igal iteratsioonil maksimeerib  $\theta^{(l)}$  Viterbi joonduse  $v^{(l)}$  ja vaatluste  $y$  ühistõepära:

$$\theta^{(l)} = \operatorname{argmax}_{\theta} P(v^{(l)}, y|\theta).$$

*Tõestus.* Et  $P(v^{(l)}, y|\theta) = P(v^{(l)}|\theta) \cdot P(y|v^{(l)}, \theta)$ , saame

$$\begin{aligned} \ln P(v^{(l)}, y|\theta) &= \ln P(v^{(l)}|\theta) + \ln P(y|v^{(l)}, \theta) = \\ &= \ln P(v^{(l)}|A, B) + \ln P(y|v^{(l)}, A, B) = \\ &= \ln P(v^{(l)}|A) + \ln P(y|v^{(l)}, B), \end{aligned}$$

kus viimane võrdus kehtib, kuna  $P(v^{(l)}|A, B)$  leidmisel ei ole meil infot vaatluste kohta ning joonduse  $v^{(l)}$  tõenäosuse leidmiseks peame teadma vaid ülemineku-tõenäosusi. Analoogiliselt on meil  $P(y|v^{(l)}, A, B)$  leidmisel  $v^{(l)}$  fikseeritud, seega  $y$  tõenäosuse leidmiseks piisab meil teada vaid emissioonitõenäosusi. Seega võime leida maksimumi mõlema liidetava kohta eraldi.

Esmalt leiame

$$\operatorname{argmax}_A \ln P(v^{(l)}|A).$$

Tähistame  $n_{ij}$  üleminekute arvu olekust  $i$  olekusse  $j$  joonduses  $v^{(l)}$  ehk

$$n_{ij} = \sum_{t=1}^{T-1} I_{\{(s_i, s_j)\}}(v_t^{(l)}, v_{t+1}^{(l)}),$$

tõepärafunktsiooni saame kirjutada kujul

$$L(A) = \operatorname{argmax}_A P(v^{(l)}|A) = \prod_{i=1}^N \prod_{j=1}^N a_{ij}^{n_{ij}}.$$

Siis logaritmtõepära on kujul

$$l(A) = \operatorname{argmax}_A \ln P(v^{(l)}|A) = \sum_{i=1}^N \sum_{j=1}^N n_{ij} \ln a_{ij}. \quad (3.1)$$

Üleminekumaatriksile  $A$  kehtivad kitsendused

$$\sum_{j=1}^N a_{ij} = 1, \quad a_{ij} \geq 0 \quad \forall j. \quad (3.2)$$

Kuna kitsendused (3.2) kehtivad maatriksi  $A$  iga rea kohta, siis võime tõepära (3.1) maksimeerida iga rea kohta eraldi. Seega iga maatriksi  $A$  rea

$$A_i := (A_{i1}, \dots, A_{iN})$$



puhul maksimeerime funktsiooni

$$l(A_i) = \sum_{j=1}^N n_{ij} \ln a_{ij}.$$

Lagrange kordajate meetodil, kasutades kitsendusi (3.2) saame

$$\begin{aligned} \frac{\partial}{\partial a_{ij}} \left( \sum_{j=1}^N n_{ij} \ln a_{ij} + \lambda_i \left( \sum_{j=1}^N a_{ij} - 1 \right) \right) &= 0 \\ \frac{n_{ij}}{a_{ij}} + \lambda_i &= 0 \quad \forall j \\ n_{ij} + \lambda_i \cdot a_{ij} &= 0 \quad \forall j \end{aligned} \tag{3.3}$$

Summeerime üle kõikide indeksite  $j$  ning võtame arvesse kitsenduse (3.2).

$$\begin{aligned} \sum_{k=1}^N n_{ik} + \lambda_i \sum_{k=1}^N a_{ik} &= 0 \\ \sum_{k=1}^N n_{ik} + \lambda_i &= 0 \\ \lambda_i &= - \sum_{k=1}^N n_{ik} \end{aligned} \tag{3.4}$$

Kui  $\sum_{k=1}^N n_{ik} \neq 0$ , siis asendame  $\lambda_i$  väärtused (3.4) võrrandisse (3.3), saame

$$\frac{n_{ij}}{a_{ij}} - \sum_{k=1}^N n_{ik} = 0 \quad \forall j.$$

Seega iga  $i = 1, \dots, N$  ja iga  $j = 1, \dots, N$  korral

$$a_{ij} = \frac{n_{ij}}{\sum_{k=1}^N n_{ik}}.$$

Kui  $\sum_{k=1}^N n_{ik} = 0$ , siis joonduses  $v^{(l)}$  ei ole kordagi olekut  $i$ , mistõttu suurima tõepäraga üleminekumaatriks pole üheselt määratud ning ei sõltu  $i$ -ndast reast. Järelikult jättes sel juhul  $a_{ij}$  hinnangu samaks, mis eelmisel iteratsioonil, saame ühe suurima tõepäraga üleminekumaatriksi.

Hinnangu

$$\operatorname{argmax}_B \ln P(y|v^{(l)}, B)$$

saab leida tähistades  $m_{ik}$  vaatluste  $k$  arvu olekus  $i$  joonduses  $v^{(l)}$  ehk

$$m_{ik} = \sum_{t=1}^T I_{\{(s_i, o_k)\}}(v_t^{(l)}, y_t).$$

Siis logaritmtõepära saame kirjutada kujul

$$l(B) = \operatorname{argmax}_B \ln P(y|v^{(l)}, B) = \sum_{i=1}^N \sum_{k=1}^M m_{ik} \ln b_i(k). \quad (3.5)$$

Emissioonimaatriksi  $B$  iga rea kohta kehtivad kitsendused

$$\sum_{k=1}^M b_i(k) = 1, \quad b_i(k) \geq 0 \quad \forall k. \quad (3.6)$$

Maksimeerime tõepära (3.5) emissioonimaatriksi  $B$  iga rea

$$B_i := (B_i(1), \dots, B_i(N))$$

kohta eraldi. Seega maksimeerime funktsiooni

$$l(B_i) = \sum_{k=1}^N m_{ik} \ln b_i(k).$$

Lagrange kordajate meetodil, kasutades kitsendusi (3.6) saame

$$\begin{aligned} \frac{\partial}{\partial b_i(k)} \left( \sum_{k=1}^M m_{ik} \ln b_i(k) + \lambda_i \left( \sum_{k=1}^M b_i(k) - 1 \right) \right) &= 0 \\ \frac{m_{ik}}{b_i(k)} + \lambda_i &= 0 \quad \forall k \end{aligned} \quad (3.7)$$

Korrutame võrrandi (3.7) läbi suurusega  $b_i(k)$  ning summeerime üle kõikide indeksite  $k$ , võttes arvesse kitsendused (3.6).

$$\begin{aligned} \sum_{j=1}^M m_{ij} + \lambda_i \sum_{j=1}^M b_i(j) &= 0 \\ \lambda_i &= - \sum_{j=1}^M m_{ij} \end{aligned} \quad (3.8)$$

Kui  $\sum_{j=1}^M m_{ij} \neq 0$ , siis asendame  $\lambda_i$  väärtused (3.8) võrrandisse (3.7), saame

$$\frac{m_{ik}}{b_i(k)} - \sum_{j=1}^M m_{ij} = 0 \quad \forall k.$$

Seega iga  $i = 1, \dots, N$  ja iga  $k = 1, \dots, M$  korral

$$b_i(k) = \frac{m_{ik}}{\sum_{j=1}^N m_{ij}}.$$

Kui  $\sum_{j=1}^M m_{ij} = 0$ , siis joonduses  $v^{(l)}$  ei ole kordagi olekut  $i$ , mistõttu suurima tõepäraga emissioonitõenäosuste maatriks pole üheselt määratud ning ei sõltu  $i$ -ndast reast. Järelikult jättes sel juhul  $b_i(k)$  hinnangu samaks, mis eelmisel iteratsioonil, saame ühe suurima tõepäraga emissioonitõenäosuste maatriksi. □

**Lause 3.** Viterbi treeningu iteratsioonide arv on alati lõplik.

*Tõestus.* Vaatluste jada  $y$  on fikseeritud. Defineerime funktsiooni

$$f(\theta, v) := P(Y_1 = y_1, \dots, Y_T = y_T, X_1 = v_1, \dots, X_T = v_T | \theta)$$

Olgu  $\theta^{(i)}$   $i$ -ndal iteratsioonil leitud parameetrid ning  $v^{(i)}$   $i$ -ndal iteratsioonil leitud joondus. Kuna iga  $i$  korral

$$v^{(i+1)} = \operatorname{argmax}_{z \in S^T} f(\theta^{(i)}, z),$$

siis

$$f(\theta^{(i)}, v^{(i)}) \leq f(\theta^{(i)}, v^{(i+1)}). \quad (3.9)$$

Lause 2 põhjal

$$\theta^{(i)} = \operatorname{argmax}_{\theta} P(v^{(i)}, y | \theta),$$

seega

$$f(\theta^{(i)}, v^{(i+1)}) \leq f(\theta^{(i+1)}, v^{(i+1)}). \quad (3.10)$$

Kui  $f(\theta^{(i)}, v^{(i+1)}) = f(\theta^{(i+1)}, v^{(i+1)})$ , siis  $f(\theta^{(i)}, v^{(i)}) = f(\theta^{(i)}, v^{(i+1)})$ . Kuna algoritmi parameetrite hinnang on igal iteratsioonil ühene, peab  $v^{(i)} = v^{(i+1)}$ , mistõttu  $\theta^{(i)} = \theta^{(i+1)}$ . Seega avaldise (3.10) kehtib võrdus parajasti siis, kui  $\theta^{(i)} = \theta^{(i+1)}$ . Järelikult eeldusel  $\theta^{(i)} \neq \theta^{(i+1)}$  saame (3.9) ja (3.10) põhjal

$$f(\theta^{(i)}, v^{(i)}) < f(\theta^{(i+1)}, v^{(i+1)}). \quad (3.11)$$

Veendume esmalt, et ühegi eelneva joonduse juurde tagasi ei minna.

Oletame vastuväiteliselt, et leidub  $k \geq 2$ , nii et

$$v^{(i)} = v^{(i+k)}, \quad v^{(i)} \neq v^{(i+m)} \quad \forall m \in \{1, \dots, k-1\}. \quad (3.12)$$

Võrduse (3.12) ning lause 2 põhjal

$$\theta^{(i)} = \operatorname{argmax}_{\theta} P(v^{(i)}, y | \theta) = \theta^{(i+k)}.$$

Seega sammul  $i$  ja sammul  $i + k$  saime samad parameetrite hinnangud, mistõttu

$$f(\theta^{(i)}, v^{(i)}) = f(\theta^{(i+k)}, v^{(i+k)}). \quad (3.13)$$

Eelduse (3.12) ning võrduse (3.11) põhjal

$$f(\theta^{(i)}, v^{(i)}) < f(\theta^{(i+1)}, v^{(i+1)}) < \dots < f(\theta^{(i+k)}, v^{(i+k)}),$$

mis on vastuolus võrdusega (3.13). Järelikult ühegi eelneva joonduse juurde tagasi ei minda.

Mudelil on kokku  $N$  olekut ning vaatluste jada koosneb  $T$  elemendist. Järelikult kokku on võimalikke joondusi  $N^T$ . Seega kui ühelgi sammul Viterbi joondused ei ole võrdsed, siis hiljemalt sammul  $N^T$  on kõik võimalikud joondused läbi vaadatud. Kuna ühelgi iteratsioonil eelnevate joonduste juurde tagasi ei minda, siis algoritmi iteratsioonide arv peab olema lõplik.

□

Paneme tähele, et kui Viterbi treeningu  $l$ -ndal iteratsioonil  $a_{ij} = 0$ , siis joonduses  $v^{(l)}$  ei leidunud ühtegi üleminekut olekust  $i$  olekusse  $j$ . Seega ka ühelgi järgneval iteratsioonil ei saa elemendi  $a_{ij}$  hinnang suureneda. Analoogiliselt, kui mingil iteratsioonil  $b_j(k) = 0$ , siis vastavas joonduses ei vastanud olekule  $j$  kordagi vaatlus  $k$ . Järelikult ei saa järgnevatel iteratsioonidel emissioonitõenäosus  $b_j(k)$  olla nullist suurem.

Vaatame paari konkreetset näidet, milliseid parameetrite hinnanguid Viterbi treening väljastab.

**Näide 3.1.** Olgu meie mudelis olekud  $S = \{1, 2\}$  ning neli vaatlust  $O = \{A, T, C, G\}$ . Olgu üleminekumaatriks, emissioonimaatriks ja algoleku tõenäosuste jaotus vastavalt

$$A = \begin{pmatrix} 0,2 & 0,8 \\ 0,4 & 0,6 \end{pmatrix}, B = \begin{pmatrix} 0,2 & 0,4 & 0,3 & 0,1 \\ 0,3 & 0,1 & 0,5 & 0,1 \end{pmatrix}, \pi = (0,5; 0,5).$$

Genereerime andmete põhjal 100-elemendilise vaatluste jada

$y = (\text{T, G, A, C, T, A, C, A, C, C, A, T, T, C, G, T, C, C, C, A, G, A, C, C, C, C, A, G, C, T, T, C, C, C, A, A, T, A, C, G, T, T, G, G, T, C, C, C, T, A, C, T, C, C, C, C, T, C, A, G, G, T, T, C, C, T, C, A, A, T, G, C, A, T, C, A, A, T, C, G, C, T, C, C, T, C, C, T, A, C, T, C, T, T, C, A, G, T, C}).$

$$x = (2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, \\ 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, \\ 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2).$$
$$A = \begin{pmatrix} 0,1852 & 0,8148 \\ 0,2917 & 0,7083 \end{pmatrix}, B = \begin{pmatrix} 0,0000 & 1 & 0,0000 & 0,0000 \\ 0,2603 & 0 & 0,5753 & 0,1644 \end{pmatrix}$$

**Näide 3.2.** Vaatleme endiselt olekuid  $S = \{1, 2\}$  ning nelja vaatlust  $O = \{A, T, C, G\}$ . Olgu üleminekumaatriks, emissioonimaatriks, algoleku tõenäosused ning vaatluste jada samad, mis näites 3.1.

$$A = \begin{pmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}, B = \begin{pmatrix} 0,1 & 0,4 & 0,2 & 0,3 \\ 0,3 & 0,1 & 0,2 & 0,4 \end{pmatrix}, \pi = (0,5; 0,5).$$
$$A = \begin{pmatrix} 0,7059 & 0,2941 \\ 0,6452 & 0,3548 \end{pmatrix}, B = \begin{pmatrix} 0,0000 & 0,3913 & 0,6087 & 0,0000 \\ 0,6129 & 0,0000 & 0,0000 & 0,3871 \end{pmatrix}$$

Kuna Viterbi treening kasutab igal iteratsioonil parameetrite hindamiseks vaid ühte joondust, siis võivad Viterbi treeningu väljundiks olla hinnangud, mis on tegelikest väärtustest väga kaugel. Selle asemel, et kasutada vaid suurima tõenäo-

susega joondust, võime arvesse võtta ka teised võimalikud joondused ning leida nende põhjal parameetrite suurima tõepära hinnangud.

### 3.2 EM algoritm

EM algoritm on iteratiivne algoritm leidmaks suurima tõepära hinnangut mudeli parameetritele varjatud andmete korral. Iga iteratsioon koosneb kahest sammust: E-sammul (*expectation step*) leitakse tõepärafunktsioon, kasutades eelmisel sammul leitud parameetrite hinnanguid. M-sammul (*maximization step*) maksimeeritaks tõepärafunktsioon eeldusel, et varjatud andmed on teada. Varjatud andmete tegelike väärtuste asemel kasutatakse E-sammul leitud hinnangut. Varjatud Markovi mudeli korral nimetatakse EM algoritmi ka Baum-Welch algoritmiks. Baum-Welch algoritm baseerub kahe funktsiooni arvutamisele, mida tuntakse *forward* ja *backward* valemitena (vt ptk 1.3).

Vaatleme abifunktsiooni

$$Q(\lambda|\lambda^{(c)}) := \sum_x P(x|y, \lambda^{(c)}) \cdot \ln P(x, y|\lambda), \quad (3.14)$$

kus  $\lambda^{(c)}$  tähistab esialgseid parameetrite hinnanguid. Näitame, et (3.14) maksimeerimine üle  $\lambda$  ei vähenda vaatlusandmete nägemise tõepära [1, 5].

**Lause 4.** Kui  $\lambda^{(c+1)} = \operatorname{argmax}_{\lambda} Q(\lambda|\lambda^{(c)})$ , siis

$$P(y|\lambda^{(c+1)}) \geq P(y|\lambda^{(c)}).$$

*Tõestus.* Tingliku tõenäosuse definitsiooni põhjal

$$\frac{P(x, y|\lambda)}{P(y|\lambda)} = P(x|y, \lambda).$$

Võtame eelmisest võrdusest logaritmi ning seejärel korrutame avaldist suurusega  $P(x|y, \lambda^{(c)})$ .

$$\ln P(x, y|\lambda) - \ln P(y|\lambda) = \ln P(x|y, \lambda)$$

$$\ln P(x, y|\lambda) \cdot P(x|y, \lambda^{(c)}) - \ln P(y|\lambda) \cdot P(x|y, \lambda^{(c)}) = \ln P(x|y, \lambda) \cdot P(x|y, \lambda^{(c)})$$

Summeerime üle kõikide joonduste  $x$ :

$$\begin{aligned} & \sum_x \ln P(x, y|\lambda) \cdot P(x|y, \lambda^{(c)}) - \ln P(y|\lambda) \sum_x P(x|y, \lambda^{(c)}) \\ &= \sum_x \ln P(x|y, \lambda) \cdot P(x|y, \lambda^{(c)}) \end{aligned}$$

Kuna  $\sum_x P(x|y, \lambda^{(c)}) = 1$ , siis jääb järele võrdus

$$\sum_x \ln P(x, y|\lambda) \cdot P(x|y, \lambda^{(c)}) - \ln P(y|\lambda) = \sum_x \ln P(x|y, \lambda) \cdot P(x|y, \lambda^{(c)}).$$

Paneme tähele, et eelneva võrduse saame kirja panna kui

$$Q(\lambda|\lambda^{(c)}) - \sum_x \ln P(x|y, \lambda) \cdot P(x|y, \lambda^{(c)}) = \ln P(y|\lambda). \quad (3.15)$$

Kuna võrdus (3.15) kehtib iga  $\lambda$  korral, siis peab see kehtima ka  $\lambda^{(c)}$  ja  $\lambda^{(c+1)}$  korral, seega

$$Q(\lambda^{(c)}|\lambda^{(c)}) - \sum_x \ln P(x|y, \lambda^{(c)}) \cdot P(x|y, \lambda^{(c)}) = \ln P(y|\lambda^{(c)})$$

$$Q(\lambda^{(c+1)}|\lambda^{(c)}) - \sum_x \ln P(x|y, \lambda^{(c+1)}) \cdot P(x|y, \lambda^{(c)}) = \ln P(y|\lambda^{(c+1)})$$

Väite tõestuseks piisab näidata, et

$$Q(\lambda^{(c+1)}|\lambda^{(c)}) \geq Q(\lambda^{(c)}|\lambda^{(c)}) \quad (3.16)$$

ja

$$\sum_x \ln P(x|y, \lambda^{(c+1)}) \cdot P(x|y, \lambda^{(c)}) \leq \sum_x \ln P(x|y, \lambda^{(c)}) \cdot P(x|y, \lambda^{(c)}). \quad (3.17)$$

Võrratus (3.16) kehtib tänu eeldusele  $\lambda^{(c+1)} = \operatorname{argmax}_{\lambda} Q(\lambda|\lambda^{(c)})$ .

Võrratuse (3.17) tõestamiseks veendume, et

$$\begin{aligned} & \sum_x \left( \ln P(x|y, \lambda^{(c)}) - \ln P(x|y, \lambda^{(c+1)}) \right) P(x|y, \lambda^{(c)}) \geq 0 : \\ & \sum_x \left( \ln P(x|y, \lambda^{(c)}) - \ln P(x|y, \lambda^{(c+1)}) \right) P(x|y, \lambda^{(c)}) \\ &= \sum_x - \left( \ln P(x|y, \lambda^{(c+1)}) - \ln P(x|y, \lambda^{(c)}) \right) P(x|y, \lambda^{(c)}) \\ &= \sum_x - \ln \left( \frac{P(x|y, \lambda^{(c+1)})}{P(x|y, \lambda^{(c)})} \right) P(x|y, \lambda^{(c)}). \end{aligned}$$

Kuna  $-\ln$  on kumer funktsioon, siis Jenseni võrratuse põhjal

$$\begin{aligned} \sum_x - \ln \left( \frac{P(x|y, \lambda^{(c+1)})}{P(x|y, \lambda^{(c)})} \right) P(x|y, \lambda^{(c)}) &\geq - \ln \sum_x \frac{P(x|y, \lambda^{(c+1)})}{P(x|y, \lambda^{(c)})} P(x|y, \lambda^{(c)}) \\ &= - \ln \sum_x P(x|y, \lambda^{(c+1)}) = 0. \end{aligned}$$

□

Nägime, et  $Q$  maksimeerimine ei kahanda vaatlusandmete nägemise tõepära. Saab näidata (vt nt [6]), et EM algoritm koondub lokaalseks maksimumiks, kuid ei pruugi anda globaalset maksimumi. Asjaolu, kas algoritm koondub lokaalseks või globaalseks maksimumiks, sõltub sisendparameetrite valikust.

Järgnevalt vaatame, kuidas leida EM algoritmi abil hinnangud HMM parameetritele. Q-funktsiooni kuju tuletamise võib leida [7].

Fikseeritud  $x$  ja  $\lambda$  korral

$$P(y|x, \lambda) = \prod_{t=1}^T P(y_t|x_t) = b_{x_1}(y_1) \cdot \dots \cdot b_{x_T}(y_T) = \prod_{t=1}^T b_{x_t}(y_t)$$

$$P(x|\lambda) = \pi_{x_1} \cdot a_{x_1x_2} \cdot a_{x_2x_3} \cdot \dots \cdot a_{x_{T-1},x_T} = \pi_{x_1} \prod_{t=1}^{T-1} a_{x_t,x_{t+1}}$$

Seega

$$P(x, y|\lambda) = P(y|x, \lambda) \cdot P(x|\lambda) = \pi_{x_1} \prod_{t=1}^{T-1} a_{x_t,x_{t+1}} \prod_{t=1}^T b_{x_t}(y_t)$$

$$\ln P(x, y|\lambda) = \ln \pi_{x_1} + \sum_{t=1}^{T-1} \ln a_{x_t,x_{t+1}} + \sum_{t=1}^T \ln b_{x_t}(y_t)$$

Funktsioon (3.14) saab kuju

$$Q(\lambda|\lambda^{(c)}) = \sum_x \ln \pi_{x_1} \cdot P(x|y, \lambda^{(c)}) + \sum_x \sum_{t=1}^{T-1} \ln a_{x_t,x_{t+1}} \cdot P(x|y, \lambda^{(c)})$$

$$+ \sum_x \sum_{t=1}^T \ln b_{x_t}(y_t) \cdot P(x|y, \lambda^{(c)}).$$

Kuna optimeeritavad parameetrid saime kirja panna üksteisest sõltumatult, siis võime parameetrite maksimeerimisel maksimeerida kõiki parameetreid eraldi.

Esmalt leiame hinnangu algtõenäosuste jaotusele  $\pi$ .

$$\hat{\pi} = \operatorname{argmax}_{\pi} \sum_x \ln \pi_{x_1} \cdot P(x|y, \lambda^{(c)}) \quad (3.18)$$

Kuna algtõenäosuste leidmisel huvitab meid vaid joonduse  $x$  esimene väärtus, siis võime võrduse (3.18) kirjutada kujul

$$\hat{\pi} = \operatorname{argmax}_{\pi} \sum_{i=1}^N \ln \pi_i \cdot P(X_1 = i|y, \lambda^{(c)}).$$



Algtõenäosuste jaotusele kehtivad kitsendused

$$\sum_{i=1}^N \pi_i = 1, \quad \pi_i \geq 0 \quad \forall i. \quad (3.19)$$

Lagrange kordajate meetodil, kasutades kitsendusi (3.19) saame

$$\begin{aligned} \frac{\partial}{\partial \pi_i} \left( \sum_{i=1}^N \ln \pi_i \cdot P(X_1 = i|y, \lambda^{(c)}) + \mu \left( \sum_{i=1}^N \pi_i - 1 \right) \right) &= 0 \\ \frac{P(X_1 = i|y, \lambda^{(c)})}{\pi_i} + \mu &= 0 \quad \forall i \\ P(X_1 = i|y, \lambda^{(c)}) &= -\mu \cdot \pi_i \quad \forall i \end{aligned} \quad (3.20)$$

Summeerime üle kõikide indeksite  $i$  ning avaldame  $\mu$ , võttes arvesse kitsendused (3.19).

$$\begin{aligned} \sum_{j=1}^N P(X_1 = j|y, \lambda^{(c)}) &= -\mu \sum_{j=1}^N \pi_j \\ \mu &= -1 \end{aligned}$$

Asendame  $\mu$  väärtuse võrrandisse (3.20).

$$\pi_i = P(X_1 = i|y, \lambda^{(c)}) \quad \forall i$$

Järgmisena leiame hinnangu üleminekumaatriksile  $A$ .

$$\hat{A} = \operatorname{argmax}_A \sum_x \sum_{t=1}^{T-1} \ln a_{x_t, x_{t+1}} \cdot P(x|y, \lambda^{(c)}) \quad (3.21)$$

Kuna igal sammul huvitavad meid ainult elemendid  $x_t$  ja  $x_{t+1}$ , siis võime võrduse (3.21) kirjutada kujul

$$\hat{A} = \operatorname{argmax}_A \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} \ln a_{ij} \cdot P(X_t = i, X_{t+1} = j|y, \lambda^{(c)}).$$

Üleminekumaatriksi igale reale kehtivad kitsendused

$$\sum_{j=1}^N a_{ij} = 1, \quad a_{ij} \geq 0 \quad \forall j. \quad (3.22)$$

Leiame hinnangu (3.21) Lagrange kordajate meetodil eraldi iga rea kohta, kasutades kitsendusi (3.22):

$$\frac{\partial}{\partial a_{ij}} \left( \sum_{j=1}^N \sum_{t=1}^{T-1} \ln a_{ij} \cdot P(X_t = i, X_{t+1} = j|y, \lambda^{(c)}) + \mu_i \left( \sum_{j=1}^N a_{ij} - 1 \right) \right) = 0$$

$$\begin{aligned} \sum_{t=1}^{T-1} \frac{P(X_t = i, X_{t+1} = j|y, \lambda^{(c)})}{a_{ij}} + \mu_i &= 0 \quad \forall j \\ \sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = j|y, \lambda^{(c)}) &= -\mu_i \cdot a_{ij} \quad \forall j \end{aligned} \quad (3.23)$$

Summeerime üle kõikide indeksite  $j$  ning avaldame  $\mu_i$ , võttes arvesse kitsendused (3.22).

$$\begin{aligned} \sum_{k=1}^N \sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = k|y, \lambda^{(c)}) &= -\mu_i \sum_{k=1}^N a_{ik} \\ \mu_i &= - \sum_{t=1}^{T-1} P(X_t = i|y, \lambda^{(c)}) \end{aligned}$$

Asendame  $\mu_i$  väärtuse võrrandisse (3.23) ning avaldame  $a_{ij}$ .

$$\begin{aligned} \sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = j|y, \lambda^{(c)}) &= \sum_{t=1}^{T-1} P(X_t = i|y, \lambda^{(c)}) \cdot a_{ij} \quad \forall j \\ a_{ij} &= \frac{\sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = j|y, \lambda^{(c)})}{\sum_{t=1}^{T-1} P(X_t = i|y, \lambda^{(c)})} \quad \forall j \end{aligned}$$

Viimasena leiame hinnangu emissioonitõenäosuste maatriksile  $B$ .

$$\hat{B} = \operatorname{argmax}_B \sum_x \sum_{t=1}^T \ln b_{x_t}(y_t) \cdot P(x|y, \lambda^{(c)}) \quad (3.24)$$

Kuna igal sammul huvitab meid, milline element emiteeriti olekust  $x_t$ , siis võime võrduse (3.24) kirjutada kujul

$$\hat{B} = \operatorname{argmax}_B \sum_{i=1}^N \sum_{t=1}^T \ln b_i(y_t) \cdot P(X_t = i|y, \lambda^{(c)}).$$

Emissioonimaatriksi igale reale kehtivad kitsendused

$$\sum_{k=1}^M b_i(k) = 1, \quad b_i(k) \geq 0 \quad \forall k. \quad (3.25)$$

Leiame hinnangu (3.24) Lagrange kordajate meetodil eraldi iga rea kohta, kasutades kitsendusi (3.25):

$$\frac{\partial}{\partial b_i(k)} \left( \sum_{i=1}^N \sum_{t=1}^T \ln b_i(y_t) \cdot P(X_t = i|y, \lambda^{(c)}) + \mu_i \left( \sum_{k=1}^M b_i(k) - 1 \right) \right) = 0$$

$$\begin{aligned} \sum_{t=1}^T \frac{P(X_t = i|y, \lambda^{(c)})I_{y_t}(o_k)}{b_i(k)} + \mu_i &= 0 \quad \forall k \\ \sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})I_{y_t}(o_k) &= -\mu_i b_i(k) \quad \forall k \end{aligned} \quad (3.26)$$

Summeerime üle kõikide indeksite  $k$  ning avaldame  $\mu_i$ , võttes arvesse kitsendused (3.25).

$$\begin{aligned} \sum_{j=1}^M \sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})I_{y_t}(o_j) &= -\mu_i \sum_{j=1}^M b_i(j) \\ \mu_i &= - \sum_{t=1}^T P(X_t = i|y, \lambda^{(c)}) \end{aligned}$$

Asendame  $\mu_i$  väärtuse võrrandisse (3.26) ning avaldame  $b_i(k)$ .

$$\begin{aligned} \sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})I_{y_t}(o_k) &= \sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})b_i(k) \quad \forall k \\ b_i(k) &= \frac{\sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})I_{y_t}(o_k)}{\sum_{t=1}^T P(X_t = i|y, \lambda^{(c)})} \quad \forall k \end{aligned}$$

Leitud hinnanguid saame kirjeldada *backward* ja *forward* muutujate kaudu. Defineerime suuruse  $\gamma_t(i)$ :

$$\gamma_t(i) = P(X_t = i|y, \lambda).$$

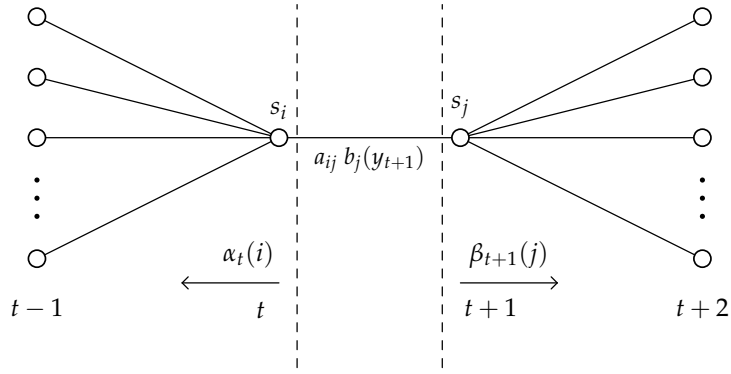
Peatüki 2.3 tulemuse põhjal

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)}.$$

Järgmisena defineerime suuruse  $\xi_t(i, j)$ :

$$\begin{aligned} \xi_t(i, j) &= P(X_t = i, X_{t+1} = j|y, \lambda) = \frac{P(X_t = i, X_{t+1} = j, y|\lambda)}{P(y|\lambda)} \\ &= \frac{\alpha_t(i)a_{ij}b_t(i)\beta_{t+1}(j)}{P(y|\lambda)} = \frac{\alpha_t(i)a_{ij}b_t(i)\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_t(i)\beta_{t+1}(j)} \end{aligned}$$

Joonis 8 illustreerib  $\xi_t(i, j)$  leidmiseks vajalikke samme.



Joonis 8: Sammud  $\xi_t(i, j)$  leidmiseks (joonis tehtud [2] põhjal)

Seega kasutades muutujaid  $\gamma_t(i)$  ja  $\xi_t(i, j)$  saame

$$\begin{cases} \pi_i = P(X_1 = i | y, \lambda^{(c)}) = \gamma_1(i) \\ a_{ij} = \frac{\sum_{t=1}^{T-1} P(X_t = i, X_{t+1} = j | y, \lambda^{(c)})}{\sum_{t=1}^{T-1} P(X_t = i | y, \lambda^{(c)})} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ b_i(k) = \frac{\sum_{t=1}^T P(X_t = i | y, \lambda^{(c)}) I_{y_t}(o_k)}{\sum_{t=1}^T P(X_t = i | y, \lambda^{(c)})} = \frac{\sum_{t=1}^T \gamma_t(i) I_{y_t}(o_k)}{\sum_{t=1}^T \gamma_t(i)} \end{cases}$$

**Näide 3.3.** Teeme näited 3.1 ja 3.2 läbi EM algoritmi kasutades. Kõigepealt anname sisendiks tegelikud parameetrite väärtused. Teeme 130 iteratsiooni, hinnangute vahed väiksemad kui 0,001 ning väljundiks on järgmised väärtused:

$$\pi = (1, 0), \quad A = \begin{pmatrix} 0,2712 & 0,7288 \\ 0,4647 & 0,5353 \end{pmatrix}, \quad B = \begin{pmatrix} 0,3175 & 0,6815 & 0,0010 & 0,0000 \\ 0,1071 & 0,0026 & 0,6923 & 0,1980 \end{pmatrix}.$$

EM algoritm andis üleminekumaatriksile üsna lähedase hinnangu, võrreldes tegeliku maatriksiga. Emissioonimaatriksi hinnang erineb tegelikust väärtusest natuke rohkem, kuid näeme, et saime paremad hinnangud kui Viterbi treeningu puhul. See-eest kuigi tegeliku algtõenäosuste jaotuse põhjal võisid olekud 1 ja 2 esineda sama tõenäosusega, siis EM algoritmi hinnangu põhjal saab algolekuks olla vaid 1.

Anname nüüd algväärtusteks ette parameetrid, nagu näites 3.2. Itereerime taas 130 korda, hinnangute vahed on väiksemad kui 0,001 ning saame parameetritele järgmised hinnangud:

$$\pi = (1, 0), \quad A = \begin{pmatrix} 0,2733 & 0,7267 \\ 0,4675 & 0,5325 \end{pmatrix}, \quad B = \begin{pmatrix} 0,3152 & 0,6802 & 0,0046 & 0,0000 \\ 0,1079 & 0,0011 & 0,6923 & 0,1987 \end{pmatrix}.$$

Näeme, et mõlemate sisendandmete korral saame parameetritele üsna lähedased hinnangud. Seega antud näite korral koondub EM algoritm mõlemate sisendandmete korral samadeks väärtusteks.

### 3.3 PMAP treening ning hindamismeetodite võrdlus

Vaatleme modifikatsiooni Viterbi treeningust, kus igal iteratsioonil leiame uue joonduse Viterbi algoritmi asemel PMAP algoritmiga. Kuna siin me enam ei leia igal iteratsioonil suurima tõenäosusega joondust, siis ei pruugi igal iteratsioonil kasvada joonduse ning vaatluste ühistõepära ning me ei pruugi saada ka suurima tõepära hinnangut. Nagu peatükis 2.2 veendusime, võib PMAP joonduses esineda keelatud üleminekuid, seega ei pruugi ka saadavate parameetrite hinnangud olla valiidsed. Kui me näiteks teame, et mingi ülemineku- või emissioonitõenäosus on võrdne nulliga, siis võib PMAP treening anda hinnangu, kus vastav tõenäosus on nullist suurem. Kuna ka iteratsioonide arv ei pruugi olla lõplik, siis algoritm lõpetab töö, kui  $u^{(i+1)} = u^{(i)}$  või kui tehtud rohkem kui 100 iteratsiooni.

**Näide 3.4.** Vaatleme olekuid  $S = \{1, 2\}$  ning nelja vaatlust  $O = \{A, T, C, G\}$ . Olgu üleminekumaatriks, emissioonimaatriks, algoleku tõenäosused järgmised:

$$A = \begin{pmatrix} 0,2 & 0,8 \\ 0,8 & 0,2 \end{pmatrix}, B = \begin{pmatrix} 0,6 & 0,0 & 0,5 & 0,3 \\ 0,4 & 0,2 & 0,3 & 0,1 \end{pmatrix}, \pi = (0,5; 0,5).$$

Genereerime 100-elementilise vaatluste jada:

$y = (T, A, T, A, C, A, C, A, C, A, T, A, T, A, G, T, A, T, C, A, C, A, A, A, C, G, A, T, G, G, A, A, A, A, A, C, T, C, C, C, T, C, A, A, A, T, C, A, C, A, T, T, C, T, A, A, A, T, C, G, C, A, A, T, A, A, T, A, A, C, A, A, A, T, A, G, G, A, A, T, C, T, A, T, A, G, C, T, A, T, C, C, C, T, G, A, G, C, G, G).$

Hindame parameetrite väärtusi nii Viterbi treeningu, EM algoritmi kui ka PMAP treeningu abil. Anname kõikidele meetoditele ette järgmised alglähendid:

$$A = \begin{pmatrix} 0,8 & 0,2 \\ 0,6 & 0,4 \end{pmatrix}, B = \begin{pmatrix} 0,2 & 0,0 & 0,5 & 0,3 \\ 0,4 & 0,2 & 0,3 & 0,1 \end{pmatrix}, \pi = (0,5; 0,5).$$

	A	B
Tegelik	$\begin{pmatrix} 0,2 & 0,8 \\ 0,8 & 0,2 \end{pmatrix}$	$\begin{pmatrix} 0,6 & 0,0 & 0,3 & 0,1 \\ 0,2 & 0,5 & 0,1 & 0,2 \end{pmatrix}$
Viterbi	$\begin{pmatrix} 0,7237 & 0,2763 \\ 0,9565 & 0,0435 \end{pmatrix}$	$\begin{pmatrix} 0,5455 & 0,0000 & 0,2987 & 0,1558 \\ 0,0000 & 1,0000 & 0,0000 & 0,0000 \end{pmatrix}$
EM	$\begin{pmatrix} 0,1653 & 0,8347 \\ 0,8717 & 0,1283 \end{pmatrix}$	$\begin{pmatrix} 0,5310 & 0,0000 & 0,3896 & 0,0794 \\ 0,3052 & 0,4678 & 0,0650 & 0,1620 \end{pmatrix}$
PMAP	$\begin{pmatrix} 0,3824 & 0,6176 \\ 0,3385 & 0,6615 \end{pmatrix}$	$\begin{pmatrix} 0,0000 & 0,0000 & 0,6571 & 0,3429 \\ 0,6462 & 0,3538 & 0,0000 & 0,0000 \end{pmatrix}$

Tabel 2: Parameetrite hinnangud erinevate meetodite korral

Tabelist 2 näeme tegelikke parameetreid ning kolme erineva treeningu väljundeid. Uurime, milline kolmest meetodist andis parimad hinnangud. Tabelit vaadates on raske otse pakkuda, milline hinnang oli parim ja milline halvim. Võrdleme meetodeid nelja erineva kriteeriumi kaudu.

Esimese võimalusena leiame, millised hinnangud on absoluutväärtuselt kõige lähedasemad tegelikele väärtustele. Selleks leiame iga meetodi korral

$$\sum_{i,j} |a_{ij} - \hat{a}_{ij}| + \sum_{i,k} |b_i(k) - b_i(\hat{k})|.$$

Järgmisena uurime, millise meetodi väljundi tõepära oli suurim. Selleks leiame

$$\ln P(y|\hat{\theta}).$$

Viimasena vaatame joonduse ja vaatluste ühistõepära. Varjatud joondust hindame nii Viterbi kui ka PMAP algoritmi abil. Seega leiame

$$\ln P(y, v(\hat{\theta})|\hat{\theta}),$$

$$\ln P(y, u(\hat{\theta})|\hat{\theta}).$$

	Viterbi	EM	PMAP
$\sum_{i,j}  a_{ij} - \hat{a}_{ij}  + \sum_{i,k}  b_i(k) - \hat{b}_i(k) $	2,472	0,6024	3,3892
$\ln P(y \hat{\theta})$	-125,1597	-123,606	-129,6493
$\ln P(y, v(\hat{\theta}) \hat{\theta})$	-125,1597	-134,6378	-129,6493
$\ln P(y, u(\hat{\theta}) \hat{\theta})$	-125,1597	-136,7119	-129,6493

Tabel 3: Viterbi treeningu, EM algoritmi ja PMAP treeningu võrdlus

Tabelist 3 näeme, et absoluutväärtuste vahest oli EM algoritmi hinnang kõige lähemal tegelikele parameetritele ning PMAP treening andis selle näite korral kõige kaugema hinnangu. Kuna EM algoritm leiab lokaalselt suurima tõepära hinnangu, siis on loomulik, et EM andis suurema tõepäraga hinnangu, kui teised meetodid. Samuti teame, et Viterbi treening maksimeerib Viterbi joonduse ja vaatluste ühistõepära. Tabelist 3 näeme, et Viterbi treening andiski vastavale tõenäosusele kõige kõrgema väärtuse. Antud näite korral saime nii Viterbi kui ka PMAP treeningu korral ühesugused hinnangud tõepärale ning mõlemale ühistõepärale. Kokkuvõttes andis antud näite korral EM algoritm kõige paremad hinnangud, sest saime kõige suurema tõepäraga parameetrid ning hinnangud olid kõige lähemal tegelikele väärtustele.

**Näide 3.5.** Uurime, kuidas sõltuvad erinevate meetodite korral parameetrite hinnangud alglähenditest vaatluste arvu  $T$  suurenedes. Valime 17 alglähendit (vt lisa 2), sealhulgas anname ette ka tegelikud parameetrite väärtused  $\theta_0$  ning leiame hinnangud  $\hat{\theta}_i$  nii EM algoritmiga, Viterbi treeninguga kui ka PMAP treeninguga.  $\theta$  väärtused üritame valida võimalikult erinevad.

Leiame iga meetodi korral parimate lõplähendite hulga, selleks defineerime EM algoritmi korral

$$I_T := \{i : P(y^T|\hat{\theta}_i) = \max_j P(y^T|\hat{\theta}_j)\},$$

Viterbi treeningu puhul

$$I_T := \{i : P(v(\hat{\theta}_i), y^T|\hat{\theta}_i) = \max_j P(v(\hat{\theta}_j), y^T|\hat{\theta}_j)\}$$

ning PMAP treeningu korral

$$I_T := \{i : P(u(\hat{\theta}_i), y^T | \hat{\theta}_i) = \max_j P(u(\hat{\theta}_j), y^T | \hat{\theta}_j)\}.$$

Genereerime 450-elementilise<sup>1</sup> vaatluste jada ning anname algoritmidele ette  $T$  esimest elementi. Võtame  $T \in \{10, 50, 100, 200, 300, 400, 450\}$ . Esitame parimate alglähendite hulga järgnevas tabelis.

	EM	Viterbi	PMAP
$I_{10}$	$\{0, 3, 5, 6, 7, 8, 13, 14\}$	$\{15\}$	$\{0\}$
$I_{50}$	$\{0, 1, 2, 3, 5, 6, 7, 8, 9, 11, 13, 14, 16\}$	$\{11\}$	$\{11\}$
$I_{100}$	$\{0, 1, 3, 5, 6, 7, 8, 11, 13\}$	$\{11\}$	$\{11\}$
$I_{200}$	$\{0, 1, 6, 7, 8, 13\}$	$\{11\}$	$\{11\}$
$I_{300}$	$\{0, 7, 13\}$	$\{11\}$	$\{11\}$
$I_{400}$	$\{0, 8, 13\}$	$\{11\}$	$\{11\}$
$I_{450}$	$\{0, 7, 8, 13\}$	$\{11\}$	$\{11\}$

Tabel 4: Parimate parameetrite indeksid

Paneme iga meetodi korral kirja ka parimad lähendid ning hinnangu täpsuse. Täpsuse hindamiseks leiame, kui kaugel olid hinnangud tegelikest väärtustest, st arvutame

$$\sum_{i,j} |a_{ij} - \hat{a}_{ij}| + \sum_{i,k} |b_i(k) - b_i(\hat{k})|.$$

---

<sup>1</sup>Kuna kirjutatud programmides on *forward* ja *backward* muutujad skaleerimata, siis suuremate  $T$  väärtuste korral läksid tõenäosused üldiselt nii väikseks, et kirjutatud programmid ei suutnud vastavaid tõenäosusi arvutada.



	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 1,000 & 0,000 \\ 0,288 & 0,712 \end{pmatrix}$	$\begin{pmatrix} 0,459 & 0,306 & 0,000 & 0,235 \\ 0,000 & 0,000 & 0,288 & 0,712 \end{pmatrix}$	1,93
Viterbi	$\begin{pmatrix} 0,0 & 1,0 \\ 0,8 & 0,2 \end{pmatrix}$	$\begin{pmatrix} 0,250 & 0,500 & 0,250 & 0,000 \\ 0,333 & 0,000 & 0,000 & 0,667 \end{pmatrix}$	4,7
PMAP	$\begin{pmatrix} 1,000 & 0,000 \\ 0,333 & 0,667 \end{pmatrix}$	$\begin{pmatrix} 0,428 & 0,286 & 0,000 & 0,286 \\ 0,000 & 0,000 & 0,333 & 0,667 \end{pmatrix}$	1,812

Tabel 5: Parameetrite hinnangud  $T = 10$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,957 & 0,043 \\ 0,112 & 0,888 \end{pmatrix}$	$\begin{pmatrix} 0,489 & 0,201 & 0,310 & 0,000 \\ 0,253 & 0,252 & 0,071 & 0,424 \end{pmatrix}$	2,368
Viterbi, PMAP	$\begin{pmatrix} 0,250 & 0,750 \\ 0,122 & 0,878 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,476 & 0,262 & 0,262 & 0,000 \end{pmatrix}$	4,432

Tabel 6: Parameetrite hinnangud  $T = 50$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,979 & 0,021 \\ 0,158 & 0,842 \end{pmatrix}$	$\begin{pmatrix} 0,545 & 0,154 & 0,248 & 0,053 \\ 0,204 & 0,255 & 0,076 & 0,465 \end{pmatrix}$	2,276
Viterbi, PMAP	$\begin{pmatrix} 0,182 & 0,818 \\ 0,102 & 0,898 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,557 & 0,193 & 0,250 & 0,000 \end{pmatrix}$	4,632

Tabel 7: Parameetrite hinnangud  $T = 100$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,967 & 0,033 \\ 0,373 & 0,627 \end{pmatrix}$	$\begin{pmatrix} 0,435 & 0,230 & 0,248 & 0,087 \\ 0,000 & 0,017 & 0,428 & 0,555 \end{pmatrix}$	1,394
Viterbi, PMAP	$\begin{pmatrix} 0,192 & 0,808 \\ 0,116 & 0,884 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,454 & 0,241 & 0,305 & 0,000 \end{pmatrix}$	4,474

Tabel 8: Parameetrite hinnangud  $T = 200$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,922 & 0,078 \\ 0,415 & 0,585 \end{pmatrix}$	$\begin{pmatrix} 0,415 & 0,236 & 0,256 & 0,093 \\ 0,040 & 0,138 & 0,218 & 0,604 \end{pmatrix}$	1,394
Viterbi, PMAP	$\begin{pmatrix} 0,283 & 0,717 \\ 0,150 & 0,850 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,429 & 0,267 & 0,304 & 0,000 \end{pmatrix}$	4,226

Tabel 9: Parameetrite hinnangud  $T = 300$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,849 & 0,151 \\ 0,418 & 0,582 \end{pmatrix}$	$\begin{pmatrix} 0,441 & 0,260 & 0,167 & 0,132 \\ 0,102 & 0,091 & 0,533 & 0,274 \end{pmatrix}$	0,54
Viterbi, PMAP	$\begin{pmatrix} 0,250 & 0,750 \\ 0,151 & 0,849 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,422 & 0,259 & 0,319 & 0,000 \end{pmatrix}$	4,26

Tabel 10: Parameetrite hinnangud  $T = 400$  korral

	$\hat{A}$	$\hat{B}$	täpsus
EM	$\begin{pmatrix} 0,818 & 0,182 \\ 0,410 & 0,590 \end{pmatrix}$	$\begin{pmatrix} 0,457 & 0,274 & 0,122 & 0,147 \\ 0,137 & 0,092 & 0,561 & 0,210 \end{pmatrix}$	0,342
Viterbi, PMAP	$\begin{pmatrix} 0,240 & 0,760 \\ 0,150 & 0,850 \end{pmatrix}$	$\begin{pmatrix} 0,000 & 0,000 & 0,000 & 1,000 \\ 0,429 & 0,262 & 0,309 & 0,000 \end{pmatrix}$	4,302

Tabel 11: Parameetrite hinnangud  $T = 450$  korral

Hinnanguid leides avastasime iga meetodi korral, et leidusid alglähendid, mis andsid sama tõepära, kuid olekud olid vahetuses. Näiteks algparameetreid vaadates märkame, et  $\theta_2$  ja  $\theta_{16}$  on üksteise permutatsioonid. Seega nende parameetrite korral saame igal iteratsioonil samad joondused, kuid olekud on vahetuses. Sellised algparameetrite komplektid annavad tegelikkuses sama hinnangu. Praktikas võib meil olla lisainfot parameetrite kohta, mille põhjal valime parema hinnangu. Kuna käesolevas näites teame tegelikke parameetrite väärtusi, siis valime sellised parameetrid, mis on tegelikele lähemal.

Tabelitest 5 – 11 näeme, et  $T = 50, 100, 200, 300, 400, 450$  korral andsid Viterbi treening ja PMAP treening samad tulemused. Vaatluste arvu suurenedes hinnang ei paranenud.

Teame, et suurima tõepära hinnang on mõjus, st vaatluste arvu kasvades hinnang koondub tegelikuks parameetriks. EM algoritm lähendab suurte  $T$  väärtuste korral suurima tõepära hinnangut. EM koondub lokaalseks maksimumiks ning mida rohkem alglähendeid valida, seda suurem on tõenäosus jõuda globaalse maksimumi. Eelolevatest tabelitest näeme, et EM algoritmi hinnangu täpsus vaatluste arvu suurenedes kasvab ning vahe hakkab lähenema nullile. Samuti paneme tähele, et tabeli 4 põhjal kuulub ka algparameeter parimate lähendite hulka. Viterbi ja PMAP treeningu puhul sama tulemus ei kehti. Tabelist 4 näeme, et tegelik parameeter ei andnud suuremate  $T$  väärtuste korral parimat hinnangut.

Kõige täpsemad hinnangud andis ka selle näite puhul EM algoritm. Nägime, et vaatluste arvu suurenedes lähenevad hinnangud tegelikele väärtustele, mida Viterbi ning PMAP treeningu puhul märgata ei ole.

## Viited

- [1] Koski, T. „Hidden Markov Models for Bioinformatics“. Kluwer Academic Publishers, 2001.
- [2] Rabiner, L. R.. „A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition“. *Proceedings of the IEEE*, vol 77, no 2, 1989, pp 257–286.
- [3] Pärna, K. „Tõenäosusteooria algkursus“. Tartu ülikooli kirjastus, 2013.
- [4] Lember, J. „Tehisõpe I“ (loengukonspekt). Tartu Ülikool, 2017.
- [5] Borman, S. „The Expectation Maximization Algorithm. A short tutorial“, 2004.  
<https://pdfs.semanticscholar.org/7ba3/f7bc41e2840ece9139c40b27152c16fc99c0.pdf>  
(07.05.2019)
- [6] Wu, C.F.J. „On the convergence properties of the em algorithm“. *The Annals of Statistics*, vol 11, no 1, 1983, pp 95–103.  
<https://pdfs.semanticscholar.org/acd7/d08468e6dd7234d474860ab64d5a33cacfdc.pdf>  
(07.05.2019)
- [7] Bilmes, J. A. „A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models“. International Computer Science Institute, 1998.  
<http://melodi.ee.washington.edu/people/bilmes/mypapers/em.pdf>  
(07.05.2019)

## 4 Lisad

### 4.1 Lisa 1. Simulatsioonides kasutatud R koodid

```
# Forward muutuja
forward_muutuja <- function(O,S,PI,A,B,Y){
  N=length(S)
  Tee=length(Y)

  alfa=matrix(c(rep(NA,Tee*N)), nrow = Tee, ncol = N, byrow = TRUE,
    dimnames = list(Y,S))
  for (i in 1:N) {
    alfa[1,i]=PI[i]*B[i,Y[1]]
  }
  for (t in 1:(Tee-1)) {
    for (j in 1:N) {
      summa=sum(alfa[t,1:N]*A[1:N,j])
      alfa[t+1,j]=B[j,Y[t+1]]*summa
    }
  }
  return(alfa)
}

# Backward muutuja
backward_muutuja <- function(O,S,PI,A,B,Y){
  N=length(S)
  Tee=length(Y)

  beeta=matrix(c(rep(NA,Tee*N)), nrow = Tee, ncol = N, byrow = TRUE,
    dimnames = list(Y,S))
  for (i in 1:N) {
    beeta[Tee,i]=1
  }
  for (t in (Tee-1):1) {
    for (i in 1:N) {
      beeta[t,i]=sum(beeta[t+1,1:N]*A[i,1:N]*B[1:N,Y[t+1]])
    }
  }
  return(beeta)
}
```

```

# Gamma muuttuja
gamma_muuttuja <- function(O,S,PI,A,B,Y){
N=length(S)
Tee=length(Y)

gamma=matrix(c(rep(NA,Tee*N)), nrow = Tee, ncol = N, byrow = TRUE,
dimnames = list(Y,S))
alfa=forward_muuttuja(O,S,PI,A,B,Y)
beeta=backward_muuttuja(O,S,PI,A,B,Y)

for (t in 1:Tee) {
  nimetaja=sum(alfa[t,1:N]*beeta[t,1:N])
  for (i in 1:N) {
    lugeja=alfa[t,i]*beeta[t,i]
    gamma[t,i]=lugeja/nimetaja
  }
}
return(gamma)
}

# Ksii muuttuja
ksii_muuttuja <- function(O,S,PI,A,B,Y){
N=length(S)
Tee=length(Y)

ksii=array(0, dim = c(Tee-1,N,N))
alfa=forward_muuttuja(O,S,PI,A,B,Y)
beeta=backward_muuttuja(O,S,PI,A,B,Y)

for (t in 1:Tee-1) {
  for (i in 1:N) {
    for (j in 1:N) {
      lugeja=alfa[t,i]*beeta[t+1,j]*A[i,j]*B[j,Y[t+1]]
      nimetaja=0
      for (ii in 1:N) {
        nimetaja=nimetaja+sum(alfa[t,ii]*beeta[t+1,1:N]*A[ii,1:N]*
        B[1:N,Y[t+1]])
      }
      ksii[t,i,j]=lugeja/nimetaja
    }
  }
}
return(ksii)
}

```

```

# Viterbi algoritm
viterbi_algoritm <- function(O,S,PI,A,B,Y){
  N=length(S) # olekute arv
  Tee=length(Y) # vaatluste jada pikkus
  delta=matrix(c(rep(NA,Tee*N)), nrow = Tee , ncol = N, byrow = TRUE)
  psii=matrix(c(rep(NA,Tee*N)), nrow = Tee , ncol = N, byrow = TRUE)

  for (i in 1:N) {
    delta[1,i]=PI[i]*B[i,Y[1]]
    psii[1,i]=0
  }
  for (t in 1:(Tee-1)) {
    for (j in 1:N) {
      delta[t+1,j]=max(delta[t,1:N]*A[1:N,j])*B[j,Y[t+1]]
      psii[t,j]=which.max(delta[t,1:N]*A[1:N,j])
    }
  }
  # Leiame Viterbi joonduse
  v=c(NA)
  v[Tee]=which.max(delta[Tee,1:N])
  for (t in (Tee-1):1) {
    v[t]=psii[t,v[t+1]]
  }
  return(S[v])
}

# PMAP joondus
PMAP_algoritm <- function(O,S,PI,A,B,Y){
  N=length(S)
  Tee=length(Y)

  gamma=gamma_muutuja(O,S,PI,A,B,Y)

  q=c(rep(1:Tee))
  for (t in 1:Tee){
    q[t]=which.max(gamma[t,1:N])
  }
  return(S[q])
}

# Defineerime abifunktsioonid ülemineku ja emissioonide tõenäosuste
# leidmiseks.

# Funktsioon leiab, mitu korda järgneb vektoris elemendile a element b.
a_jargneb_b <- function(vektor,a,b){

```

```

kokku=0
for (i in 1:(length(vektor)-1)) {
  if(vektor[i]==a & vektor[i+1]==b){
    kokku=kokku+1
  }
}
return(kokku)
}

#Funksioon leiab, mitu korda järgneb vektoris elemendile a mingi element.
a_jargneb_x <- function(vektor,a){
kokku=0
if(tail(vektor, n=1)==a){
  kokku=(table(vektor)[a])-1
}
else{
  kokku=table(vektor)[a]
}
return(kokku)
}

#Funksioon leiab, mitu korda on olekute vektoris olek a ja vaatluste
#vektoris vaatlus b.
olek_a_ja_vaatlus_b <- function(vaatlused, olekud,a,b){
kokku=0
for (i in 1:length(vaatlused)) {
  if(olekud[i]==a & vaatlused[i]==b){
    kokku=kokku+1
  }
}
return(kokku)
}

# Viterbi treening
viterbi_treening <- function(O,S,PI,A,B,Y,X, loendur){
T=matrix(c(rep(0,length(S))), nrow = length(S) , ncol = length(S),
dimnames = list(S,S))
E=matrix(c(rep(0,length(O))), nrow = length(S) , ncol = length(O),
dimnames = list(S,O))
olekud=viterbi_algoritm(O,S,PI,A,B,Y)
for (olek in S) {
  for (v in O) {
    if (is.na((table(olekud))[olek])){
      E[olek, v]=B[olek, v]
    }
  }
}

```



```

        else{
            E[olek, v]=(olek_a_ja_vaatlus_b(Y, olekud, olek, v))/
                ((table(olekud))[olek])
        }
    }
}
for (olek1 in S) {
    for (olek2 in S) {
        if (is.na(a_jargneb_x(olekud,olek1))){
            T[olek1, olek2]=A[olek1, olek2]
        }
        else{
            T[olek1, olek2]=(a_jargneb_b(olekud,olek1,olek2))/
                (a_jargneb_x(olekud,olek1))
        }
    }
}
if (identical(olekud, X)) {
    vastus <- list(T,E,loendur)
    names(vastus) <- c("A","B","iteratsioone")
    return(vastus)
}
else return (viterbi_treening(O,S,PI,T,E,Y,olekud,loendur+1))
}

# EM algoritm HMM jaoks
EM_algoritm <- function(O,S,PI,A,B,Y,loendur){
N=length(S)
K=length(O)
Tee=length(Y)

alfa=forward_muutuja(O,S,PI,A,B,Y)
beeta=backward_muutuja(O,S,PI,A,B,Y)
gamma=gamma_muutuja(O,S,PI,A,B,Y)
ksii=ksii_muutuja(O,S,PI,A,B,Y)

# Uus PI väärtus
PI_uus=c(NA)
for (i in 1:N) {
    PI_uus[i]=gamma[1,i]
}

# Uus A väärtus
A_uus=matrix(c(rep(NA,N*N)), nrow = length(S), ncol = length(S),
byrow = T, dimnames = list(S,S))

```

```

for (i in 1:N) {
  for (j in 1:N) {
    A_uus[i,j]=sum(ksii[1:Tee-1,i,j])/sum(gamma[1:Tee-1,i])
  }
}

# Uus B väärtus
B_uus=matrix(c(rep(NA,N*K)), nrow = length(S), ncol = length(O),
byrow = T, dimnames = list(S,O))

iden <- function(t,k){
  if (Y[t]==0[k]) return(1)
  return(0)
}

for (i in 1:N) {
  for (k in 1:K) {
    lugeja=0
    for (t in 1:Tee) {
      lugeja=lugeja+gamma[t,i]*iden(t,k)
    }
    B_uus[i,k]=lugeja/sum(gamma[1:Tee,i])
  }
}

if(loendur>1) return(EM_algoritm(O,S,PI_uus,A_uus,B_uus, Y, loendur-1))

vastus <- list(PI_uus, A_uus, B_uus)
names(vastus) <- c("PI","A","B")
return(vastus)
}

# PMAP treening
PMAP_treening <- function(O,S,PI,A,B,Y,X, loendur){
T=matrix(c(rep(0,length(S))), nrow = length(S) , ncol = length(S),
dimnames = list(S,S))
E=matrix(c(rep(0,length(O))), nrow = length(S) , ncol = length(O),
dimnames = list(S,O))
olekud=PMAP_algoritm(O,S,PI,A,B,Y)
for (olek in S) {
  for (v in O) {
    if (is.na((table(olekud))[olek])){
      E[olek, v]=B[olek, v]
    }
    else{

```

```

        E[olek, v]=(olek_a_ja_vaatlus_b(Y, olekud, olek, v))/
        ((table(olekud))[olek])
    }
}
}
for (olek1 in S) {
  for (olek2 in S) {
    if (is.na(a_jargneb_x(olekud,olek1))){
      T[olek1, olek2]=A[olek1, olek2]
    }
    else{
      T[olek1, olek2]=(a_jargneb_b(olekud,olek1,olek2))/
      (a_jargneb_x(olekud,olek1))
    }
  }
}
if (identical(olekud, X) | loendur>100) {
  vastus <- list(T,E,loendur)
  names(vastus) <- c("A","B","iteratsioone")
  return(vastus)
}
else return (PMAP_treening(0,S,PI,T,E,Y,olekud,loendur+1))
}

# Funktsioon, mis leiab joonduse x ja vaatluste y ühistõepära.
p_xy_ting_lambda <- function(0,S,PI,A,B,X,Y){
  vastus <- PI[X[1]]
  for (t in 1:(length(Y)-1)) {
    vastus <- vastus*A[X[t],X[t+1]]
  }
  for (t in 1:(length(Y))) {
    vastus <- vastus*B[X[t],Y[t]]
  }
  return(vastus)
}

# Funktsioon, mis leiab suurima tõepära hinnangu
stp <- function(0,S,PI,A,B,Y){
  alfa=forward_muutuja(0,S,PI,A,B,Y)
  return(log(sum(alfa[length(Y),1:length(S)])))
}

```

## 4.2 Lisa 2. Alglähenditeks antud parameetrid

	$A_i$	$B_i$
$i = 0$ (tegelik)	$\begin{pmatrix} 0,8 & 0,2 \\ 0,4 & 0,6 \end{pmatrix}$	$\begin{pmatrix} 0,5 & 0,25 & 0,1 & 0,15 \\ 0,05 & 0,15 & 0,6 & 0,2 \end{pmatrix}$
$i = 1$	$\begin{pmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}$	$\begin{pmatrix} 0,4 & 0,2 & 0,3 & 0,1 \\ 0,1 & 0,3 & 0,1 & 0,5 \end{pmatrix}$
$i = 2$	$\begin{pmatrix} 0,2 & 0,8 \\ 0,3 & 0,7 \end{pmatrix}$	$\begin{pmatrix} 0,1 & 0,4 & 0,2 & 0,3 \\ 0,4 & 0,1 & 0,4 & 0,1 \end{pmatrix}$
$i = 3$	$\begin{pmatrix} 0,4 & 0,6 \\ 0,1 & 0,9 \end{pmatrix}$	$\begin{pmatrix} 0,6 & 0,2 & 0,15 & 0,05 \\ 0,3 & 0,2 & 0,1 & 0,4 \end{pmatrix}$
$i = 4$	$\begin{pmatrix} 0,5 & 0,5 \\ 0,6 & 0,4 \end{pmatrix}$	$\begin{pmatrix} 0,25 & 0,1 & 0,5 & 0,15 \\ 0,2 & 0,4 & 0,2 & 0,2 \end{pmatrix}$
$i = 5$	$\begin{pmatrix} 0,7 & 0,3 \\ 0,5 & 0,5 \end{pmatrix}$	$\begin{pmatrix} 0,3 & 0,1 & 0,4 & 0,2 \\ 0,2 & 0,3 & 0,3 & 0,2 \end{pmatrix}$
$i = 6$	$\begin{pmatrix} 0,9 & 0,1 \\ 0,5 & 0,5 \end{pmatrix}$	$\begin{pmatrix} 0,25 & 0,1 & 0,15 & 0,5 \\ 0,1 & 0,6 & 0,05 & 0,25 \end{pmatrix}$
$i = 7$	$\begin{pmatrix} 0,8 & 0,2 \\ 0,9 & 0,1 \end{pmatrix}$	$\begin{pmatrix} 0,15 & 0,25 & 0,2 & 0,4 \\ 0,05 & 0,2 & 0,7 & 0,05 \end{pmatrix}$
$i = 8$	$\begin{pmatrix} 0,4 & 0,6 \\ 0,6 & 0,4 \end{pmatrix}$	$\begin{pmatrix} 0,35 & 0,35 & 0,2 & 0,1 \\ 0,1 & 0,1 & 0,4 & 0,4 \end{pmatrix}$

$i = 9$	$\begin{pmatrix} 0,7 & 0,3 \\ 0,7 & 0,3 \end{pmatrix}$	$\begin{pmatrix} 0,15 & 0,15 & 0,35 & 0,35 \\ 0,35 & 0,35 & 0,15 & 0,15 \end{pmatrix}$
$i = 10$	$\begin{pmatrix} 0,1 & 0,9 \\ 0,8 & 0,2 \end{pmatrix}$	$\begin{pmatrix} 0,05 & 0,6 & 0,05 & 0,3 \\ 0,35 & 0,25 & 0,15 & 0,25 \end{pmatrix}$
$i = 11$	$\begin{pmatrix} 0,2 & 0,8 \\ 0,2 & 0,8 \end{pmatrix}$	$\begin{pmatrix} 0,1 & 0,2 & 0,3 & 0,4 \\ 0,4 & 0,3 & 0,2 & 0,1 \end{pmatrix}$
$i = 12$	$\begin{pmatrix} 0,25 & 0,75 \\ 0,65 & 0,35 \end{pmatrix}$	$\begin{pmatrix} 0,1 & 0,4 & 0,25 & 0,25 \\ 0,5 & 0,3 & 0,1 & 0,1 \end{pmatrix}$
$i = 13$	$\begin{pmatrix} 0,55 & 0,45 \\ 0,6 & 0,4 \end{pmatrix}$	$\begin{pmatrix} 0,4 & 0,3 & 0,15 & 0,15 \\ 0,1 & 0,2 & 0,55 & 0,15 \end{pmatrix}$
$i = 14$	$\begin{pmatrix} 0,15 & 0,85 \\ 0,25 & 0,75 \end{pmatrix}$	$\begin{pmatrix} 0,3 & 0,2 & 0,4 & 0,1 \\ 0,2 & 0,3 & 0,1 & 0,4 \end{pmatrix}$
$i = 15$	$\begin{pmatrix} 0,33 & 0,67 \\ 0,74 & 0,26 \end{pmatrix}$	$\begin{pmatrix} 0,5 & 0,2 & 0,15 & 0,15 \\ 0,2 & 0,15 & 0,15 & 0,5 \end{pmatrix}$
$i = 16$	$\begin{pmatrix} 0,7 & 0,3 \\ 0,8 & 0,2 \end{pmatrix}$	$\begin{pmatrix} 0,4 & 0,1 & 0,4 & 0,1 \\ 0,1 & 0,4 & 0,2 & 0,3 \end{pmatrix}$

Tabel 12: Näites 3.5 kasutatud algparameetrid

## **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Kristin Avans,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

„Segmenteerimine ja parameetrite hindamine varjatud Markovi mudeli korral“,

mille juhendaja on Jüri Lember,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Kristin Avans  
08.05.2019